Concurrent Assembly Mock-Up User Guide and Menu Reference

CADDS® 5 15.0

DOC38700-018

Parametric Technology Corporation

Copyright © 2007 Parametric Technology Corporation. All Rights Reserved.
User and training guides and related documentation from Parametric Technology Corporation and its subsidiary companies (collectively "PTC") is subject to the copyright laws of the United States and other countries and is provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

For Important Copyright, Trademark, Patent, and Licensing Information: For Windchill products, select About Windchill at the bottom of the product page. For InterComm products, on the Help main page, click the link for Copyright 2007. For other products, select Help > About on the main menu for the product.

UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND
This document and the software described herein are Commercial Computer Documentation and Software, pursuant to FAR 12.212(a)-(b) (OCT'95) or DFARS 227.7202-1(a) and 227.7202-3(a) (JUN'95), and are provided to the US Government under a limited commercial license only. For procurements predating the above clauses, use, duplication, or disclosure by the Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 (OCT'88) or Commercial Computer Software-Restricted Rights at FAR 52.227-19(c)(1)-(2) (JUN'87), as applicable. 02202007

Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494 USA

Table of Contents

Preface

Related Documents ____ xvii
Book Conventions ____ xviii
Window Managers and the User Interface ____ xix
Online User Documentation ____ xix
Online Command Help ____ xx
Printing Documentation ____ xx
Resources and Services ____ xxi
Documentation Comments ____ xxi

Introduction to Concurrent Assembly Mock-Up

What You Can Do With CAMU ____ 1-2
A Look at CAMU ____ 1-3
Local Data Manager Menu ____ 1-4
Concurrent Assembly Mock-Up Windows ____ 1-5
Overview of CAMU Windows ____ 1-5
The Assembly Structure ____ 1-7
Terminology ____ 1-7
Root component ____ 1-7
Component ____ 1-7
Component instance ____ 1-7
Assembly instance ____ 1-8
Component class ____ 1-8

Parent Component 1-8
Child Component 1-8
Siblings 1-8
Leaf Component 1-8
Relation Instance 1-8
Model 1-9
Adrawing 1-9

Assembly Tree Window 1-11
Shortcut Menus within the Assembly Tree Window 1-15
Assembly Tree Window Top Bar Options 1-19
Assembly Tree Window Toolbar Options 1-20

CAMU Top Bar Options 1-22

Assembly Environment Task Sets 1-23
Common Tasks 1-23
Moving to Other Environments and Task Sets 1-24
Moving to Other Environment Groups 1-27

Accessing Functions 1-28
Using an Icon to Select an Option 1-28
Property Sheets 1-28
Filling In Fields on a Property Sheet 1-29
Using Pushbuttons on Property Sheets 1-30
Processing the Property Sheet 1-31
Pushpin Icon 1-31
Using the Context-Sensitive Menu Within the Assembly Structure Window 1-31
Using Online Documentation 1-32
Commands and Command Help 1-33
Assist Lines 1-33
Using Online Command Help 1-33

Reversing a Command 1-36
Commands You Cannot Undo 1-36

Preparing to Work 1-37
Viewing Parts 1-37
Entering CAMU 1-37
Manipulating a Component That Has No Geometry 1-37

Establishing the DB Daemon ____ 1-38
Suspending a Specified Number of Models ____ 1-38
Accessing CAMU ____ 1-39
Accessing the Associative Topology Bus (ATB) Option ____ 1-40
Updating TIM Parts with Pending Regeneration State ____ 1-41
Querying the ATB Transactions ____ 1-42

Configuring Concurrent Assembly Mock-Up

Implementing CAMU and EPD ____ 2-2
.caddsrc-local File Information ____ 2-3
Configuring Multi-User CAMU ____ 2-7
Choosing a Host System for DB_DAEMON_HOST ____ 2-7
Performance Considerations ____ 2-7
Network Considerations ____ 2-8
NFS (Network File System) Prerequisites ____ 2-8
CVPATH Prerequisites ____ 2-10
Software Prerequisites ____ 2-10
Other System Considerations ____ 2-11
Sample Procedure to Configure Multi-User CAMU ____ 2-11
Example 1: ____ 2-12
Example 2: ____ 2-12
Example 3: ____ 2-12
Testing the Multi-User Configuration ____ 2-13
Troubleshooting ____ 2-14
Activating a New or an Existing Assembly ____ 2-14
Single User Environment ____ 2-15
Terminating the ASSEMBLY_DESIGN and ODB_SERVER Process ____ 2-16
Multi-User Environment ____ 2-16
ODB (Object Data Base) Administration Tool ____ 2-18
Arranging Projects Using CVPATH ____ 2-20
Example of setting multiple CVPATHs: ____ 2-20
Data Management ____ 2-21
Assembly Database Items ____ 2-21

Model Items 2-21
Locking Models 2-22
Archiving the Assembly 2-22
CADDS Drawing Files 2-22
vp_links Directory 2-23
Aborting a Command 2-23
Crash Recovery 2-24

Creating a New Assembly

Creating (Activating) a New Assembly 3-2
Using the Default Adrawing 3-2
Steps to Activate a New Assembly 3-2
Building the Assembly Structure 3-5
Typical Structuring Approaches 3-5
Guidelines for Creating the Assembly Structure 3-6
Naming Conventions 3-7
Ways You Can Add Components or Assemblies 3-7
Adding a New Component to the Structure 3-9
Options on the Property Sheet 3-11
Viewing and Positioning the Component 3-12
View and Position without Orienting 3-13
Position and Orient 3-14
Adding a Family Member of a Master Model 3-15
Multiple Occurrences of the Same Item 3-17
A Parent Component Having Multiple Instances 3-17
Adding Multiple Instances of the Same Component 3-18
Add Component Property Sheet Options Specified 3-18
Instance Names 3-19
Making a Change to Only One Instance of a Component 3-19
Relations Between Components in the Structure 3-20
Adding Existing Assemblies to Your Assembly 3-21
Reference Assemblies 3-21
Copied Assemblies 3-22

Adding a Reference Assembly _____ 3-23
Example of Referencing an Assembly _____ 3-24
Nested Reference Assemblies _____ 3-24
Example of a Nested Reference Assembly _____ 3-25
Controlling the Level of Nested Reference Assemblies _____ 3-26
Example: Open/Close Reference _____ 3-27
Using the Prefix Option _____ 3-30
Creating a New Assembly by Exporting _____ 3-31
Example: Exporting without Referencing _____ 3-32
Example: Exporting with Referencing _____ 3-33
Exporting a Reference Assembly _____ 3-35
Exporting the Active Assembly _____ 3-35

Filing an Assembly

Filing Your Changes _____ 4-2
Updating Associative Annotation _____ 4-2
Updating Active Reference Assemblies _____ 4-3
Deactivating a Reference Assembly _____ 4-3
Understanding System Messages _____ 4-3
Filing All Active Assemblies _____ 4-3
Filing and Locking Interactivity _____ 4-3
Filing Assembly Structure, Default Adrawing, and Root Node Model _ 4-4
Filing with and without Exiting _____ 4-4
Selections from the FILE Icon _____ 4-4
Filing or Quitting Selected Adrawings or Models _____ 4-6
Filing with Options _____ 4-6
Filing the Active Model Only _____ 4-7
Quitting with Options _____ 4-8

Working in an Existing Assembly

Activating the Main Assembly _____ 5-2
Viewing Model Geometry _____ 5-5

About Adrawings 5-5
The Purpose of Adrawings 5-5
Clarifying Terms 5-6
Moving Between Models and Adrawings 5-7
What You See in the Graphics Window 5-7
DDRAW 5-7
MDRAW 5-7
ERASE ENTITY Design Restriction 5-8
Moving to a New Adrawing or New Model 5-8
Moving to a Suspended Model 5-9
Moving to a Suspended Adrawing 5-9
Understanding the Suspension Buffer 5-9
Filing an Active Model That Is in History Replay Mode 5-9
Examples of Moving Between Adrawings and Models 5-10
Activating the Assembly and an Existing Adrawing 5-10
Activating a Model While Active in an Adrawing 5-11
Moving from One Active Model to Another 5-13
Moving from an Active Model to an Active Adrawing 5-17
Activating a New Adrawing 5-18
Moving from a New Adrawing to an Existing Adrawing 5-21
Visual Changes When Activating Parametric Models 5-21
Associative Mass Points 5-22
Using the Associative Mass Points Option 5-23
Using Merge Mass Point Options 5-24
Several Users Can Work Concurrently 5-27
Determining Status in Multiuser CAMU 5-27
Updating Your Session 5-27
Multiuser CAMU Example 5-28
Updating Parametric Positions After Modeling Changes 5-30
Checking Which Server Process is Attached to the Assembly 5-30
Locking Components, Attribute Groups, and Models 5-31
Locking Areas of the Structure 5-32
Determining What to Lock 5-32
Lock/Unlock Component Property Sheet 5-33

Activating a Reference Assembly ____ 5-34
Refreshing the Main Assembly ____ 5-36
Fully Constrained Assembly ____ 5-36
Underconstrained Assembly ____ 5-37
Overconstrained Assembly ____ 5-37

Parametric Positioning

Understanding Parametric Positioning ____ 6-2
Task Set Options ____ 6-2
Parametric Positioning Tools ____ 6-4
When Does Solving Occur ____ 6-4
Understanding Solving Process Order ____ 6-4
Using Positioning Constraints within Multiuser CAMU ____ 6-6
Path Locking ____ 6-6
Creating Positioning Constraints ____ 6-7
Locking ____ 6-7
Creating Relationships between Component Geometry ____ 6-8
Constraint Mapping between Cylindrical Surfaces ____ 6-8
Ambiguous Selection ____ 6-8
Using the Flip Option ____ 6-8
Entity Selection ____ 6-9
Multiple Instances of a Subassembly Class ____ 6-10
Creating a New Class to Capture Unique Positioning ____ 6-11
Hybrid Positioning ____ 6-12
Parametric and Explicit Positioning ____ 6-12
Explicit Models ____ 6-12
Using Reference Assemblies ____ 6-12
Using Constrain Component ____ 6-13
Constrain Component Property Sheet Options ____ 6-14
Changing Offset Value ____ 6-19
Deleting a Constraint ____ 6-20
Accessing Modify Position Constraints Menu ____ 6-20
Example: Deleting Invalid Constraints ____ 6-20
Color Scheme ____ 6-21
Example: Isolate Constraint Problems ____ 6-21

Sample Procedure for Creating Constraints _____ 6-22
Example Variant 1 _____ 6-28
Example Variant 2 _____ 6-30
Solving Parametric Positioning Constraints _____ 6-32
Automatic Assembly _____ 6-32
Manual Request to Assemble _____ 6-32
Base Component _____ 6-33
Designating a Base Component _____ 6-34
Redesignating a Base Component _____ 6-35
Recognizing Full Constraints Status _____ 6-35
Fully Constrained Assembly _____ 6-35
Underconstrained Assembly _____ 6-35
Overconstrained Assembly _____ 6-35
Changing Positioning Constraints _____ 6-36
Updating Positions After Modeling Change _____ 6-37
Using Pre-Revision 8.0 CADDS Models _____ 6-38
CADDSASSOC_UID _____ 6-38
Parametric Positioning and Adrawings _____ 6-39
Exploded View State _____ 6-39

Changing the Assembly Structure

Ways to Change the Assembly Structure _____ 7-2
Changes You Can Make to the Assembly Structure _____ 7-2
Changing the Appearance of the Structure _____ 7-3
Obtaining Information about the Structure _____ 7-3
Changing a Component _____ 7-4
Steps to Change a Component _____ 7-5
Changing a Component Name Can Create a New Class _____ 7-6
Removing a Component _____ 7-7
Component Instance Numbers _____ 7-8
Removing a Component by Removing the Relation _____ 7-9
Copying a Component _____ 7-12

Positioning the Copied Component ____ 7-14
Mirroring a Component ____ 7-15
Example: Mirroring a Component in the Parametric Environment by Creating a Part Database ____ 7-19
Specifying Orientation for Mirror Copied Component ____ 7-23
Changing a Reference Assembly ____ 7-25
Editing an External Assembly ____ 7-27
Understanding System Messages ____ 7-28
Changing Structure Appearance ____ 7-29
Blanking a Subassembly ____ 7-29
Compressing a Subassembly ____ 7-30
Reordering a Component ____ 7-32
Placing a Component above Another in the Main Assembly ____ 7-32
Placing a Component above Another in a Reference Assembly ____ 7-33
Placing a Component below Another Component ____ 7-34
Obtaining Information about Components ____ 7-35
Using the CAMU Status Window ____ 7-35
Using the Verify Object Function ____ 7-36
Listing Reference Assemblies ____ 7-39

Defining Assembly Geometry

Guidelines for Creating Model Geometry ____ 8-2
Working on Model Geometry ____ 8-3
A New Component and Its Associated Model ____ 8-3
Activating a Model ____ 8-3
Using a Source Model to Create a New Model ____ 8-4
Component Instances and Their Locations ____ 8-6
Locations of Components in Reference Assemblies ____ 8-6
Locating Model Space Origin for a New Model ____ 8-7
Creating A New Model without Locating Model Space Origin ____ 8-7
Pulley Assembly Structure ____ 8-7

Activating the Model Associated with the Base Component __ 8-9
Model Space Origin Location of the New Model _____ 8-9
Building the Model Geometry _____ 8-10
Specifying Model Space Origin before Creating the Model_____ 8-12
Position the Model Instance Associated with Bracket _____ 8-12
Locating MSO for New Model Associated with Wheel _____ 8-15
Viewing a Component's Associated Model Geometry _____ 8-18
Options on the Property Sheet _____ 8-19
Outline Geometry _____ 8-20
Controlling Layer Display_____ 8-22
Temporarily Blanking Viewed Model Geometry _____ 8-23
Checking part database of a Model Associated with the Component _____ 8-23
Moving a Component_____ 8-24
Using Translate, Rotate, and Orient Component _____ 8-25
Alternative Procedure for Orienting and Positioning wheel_2_____ 8-28
Six-Point Orientation Method _____ 8-30
Position Component_____ 8-30
Orientation Procedure_____ 8-32
Rotating a Component around an Axis passing through the Origin of the Part _____ 8-32
Rotating a Component around an Axis passing through a Selected Location_____ 8-33
Translating, Aligning, and Adjusting Components _____ 8-34
Orienting a Component: Work Examples_____ 8-35
Example 1: Rotating a Component at its Origin around x-Axis _____ 8-36
Example 2: Rotating a Component at its Origin around an Axis Defined by Specifying Two Locations _____ 8-37
Example 3: Rotating a Component around the y-Axis passing through a Location_____ 8-38
Example 4: Rotating a Component around an Axis Specified by Two Locations _____ 8-39
Example 5: Translating a Component by a Displacement Specified by Two Locations _____ 8-40
Clearing a Component's Inherited Orientation_____ 8-41
Positioning Tools _____ 8-42
Utilities for Positioning _____ 8-42

Highlighting the Appearance of Geometry _____ 8-44

Using Interpart Constraints

About Interpart Constraints _____ 9-2
Basic Steps to Use Constraints in an Assembly _____ 9-2
Selecting Variables from a Model to Include in Your Assembly _____ 9-4
Placing Component Variables in the Active Assembly _____ 9-7
Constraint Variable Names of Imported Components _____ 9-10
Preparing to Add Equations Between Variables _____ 9-12
When Are Equations Solved? _____ 9-12
Solving Equations Manually _____ 9-12
Changes to the Variables in the Associated Parts _____ 9-13
Establishing Relationships Between Variables _____ 9-14
In Automatic Mode _____ 9-15
In Manual Mode _____ 9-15
Changing Variable Values and Applying the Changes _____ 9-17
Changing Variable Values _____ 9-17
Applying the Changes (Automatic Mode Only) _____ 9-18
In Manual Mode _____ 9-19
Regenerating the Component _____ 9-20
Forcing a Variable to Remain Constant _____ 9-22
Deleting a Variable _____ 9-23

Adrawings and Alternate Views

Adrawings _____ 10-2
Significance of the Active Adrawing _____ 10-2
The Adrawing Provides Consistency _____ 10-3
The Default Adrawing _____ 10-3
Adrawings and the Alternative View State _____ 10-3

Options on the Property Sheet ____ 10-4
Annotating the Adrawing ____ 10-4
Obtaining Information About Adrawings ____ 10-4
Viewing Models in Adrawings ____ 10-5
Using Alternate Views in Your Design ____ 10-6
Basic Steps to Create an Alternate View ____ 10-7
The Standard View ____ 10-7
Selecting Exploded View ____ 10-9
In an Active Adrawing ____ 10-9
As You Activate an Adrawing ____ 10-9
The Exploded View ____ 10-10
Changing an Adrawing ____ 10-11
Deleting an Adrawing ____ 10-12
Saving and Restoring Viewstates ____ 10-12

Defining Nongraphic Information for the Assembly

About Attributes ____ 11-2
Attributes Task Set ____ 11-2
Overview of Procedure to Use Attributes ____ 11-3
Creating Attribute Groups ____ 11-4
Associating an Attribute Group with a Component ____ 11-5
Disassociating an Attribute Group from a Component ____ 11-6
Listing Attribute Groups ____ 11-7
Creating Attributes ____ 11-8
Floating-Point Numbers ____ 11-10
Editing Attributes ____ 11-11
Attribute Editor Window ____ 11-11
Locking Attributes and Attribute Groups ____ 11-15
Changing Attribute Properties ____ 11-16

Removing an Attribute ____ 11-17
Removing an Attribute Group ____ 11-18
Generating Output Using Supplied CVMACs ____ 11-19
Supplied CVMAC Programs ____ 11-19
Query ____ 11-19
Bill of Material ____ 11-19
Generate Assembly Structure Entities ____ 11-19
Output All Attribute Information ____ 11-20
Creating a CVMAC Executable ____ 11-20
Link and Compile CVMAC Procedures ____ 11-21
Creating CVMAC Executables ____ 11-21
Procedure ____ 11-21

Documenting the Assembly

Creating an Assembly Drawing Document ____ 12-2
Significance of the Active Adrawing ____ 12-2
Generating Hidden Line Representation (HLR) ____ 12-3
Creating an Adrawing Document ____ 12-3
Example ____ 12-4
Generating HLR Using HIDE OBJECT ____ 12-5
HLR Commands ____ 12-5
Removing Hidden Lines ____ 12-6
To Access HLR ____ 12-6
Assembly Structure Display Results ____ 12-14
Restoring Hidden Lines ____ 12-15
Updating the HLR Image ____ 12-16
Generating HLR Using CADDShade II ____ 12-17
File Structure and the Shade/Hide Picture File ____ 12-17
Preparing to Use CADDShade II ____ 12-17
Generating CADDShade II HLR ____ 12-18
Dimensioning and Annotating ____ 12-21
Inserting Dimensions and Annotation ____ 12-22
Establishing Associativity Default ____ 12-22

Inserting Dimensions _____ 12-24
Regenerating Dimensions _____ 12-25
Detailing Models in CAMU _____ 12-25
Plotting _____ 12-27
Plotting the CADDS Graphics Window _____ 12-27
Plotting the CADDShade Image _____ 12-29
Plotting the Assembly Structure _____ 12-29
Plotting the Part History Tree _____ 12-32
Glossary

Preface

Concurrent Assembly Mock-Up User Guide and Menu Reference provides instructions for section leaders, managers, and design engineers who design assemblies comprised of explicit and parametric CADDS models.

Related Documents

The following documents may be helpful as you use:

The following documents may be helpful as you use *Concurrent Assembly Mock-Up User Guide and Menu Reference*:

- *Explicit Modeling User Guide and Menu Reference*
- *Parametric Modeling User Guide and Menu Reference*
- *Parametric Modeling Best Practices*
- *Hidden Line Removal (and AEC HLR) User Guide and Menu Reference*
- *EPD.Connect User Guide*
- *Installing and Configuring EPD.Connect*
- *Installing CADDS 5*
- *Managing CADDS 5*
- *Harness Layout Menu and Technical Reference*
- *Design and Drafting User Guide and Menu Reference*
- *Using Associative Topology Bus Enabled CADDS 5i*

Book Conventions

The following table illustrates and explains conventions used in writing about CADDS applications.

Convention	Example	Explanation
Menu selections and options	List Section option, Specify Layer field	Indicates a selection you must make from a menu or property sheet or a text field that you must fill in.
User-selected graphic location	X, d_1 or P1	Marks a location or entity selection in graphic examples.

User input in CADDS text fields and on any command line	`cvaec.hd.data.param`
	
	`tar -xvf /dev/rst0`
		Enter the text in a CADDS text field or on any command line.

System output	`Binary transfer complete.`	Indicates system responses in the CADDS text window or on any command line.
Variable in user input	`tar -cvf /dev/rst0` filename	Replace the variable with an appropriate substitute; for example, replace filename with an actual file name.
Variable in text	tagname	Indicates a variable that requires an appropriate substitute when used in a real operation; for example, replace tagname with an actual tag name.
CADDS commands and modifiers	INSERT LINE TANTO	Shows CADDS commands and modifiers as they appear in the command line interface.
Text string	`"SRFGROUPA"` or `'SRFGROUPA'`	Shows text strings. You must enclose text string with single or double quotation marks.
Integer	n	Supply an integer for the *n*.
Real number	x	Supply a real number for the *x*.
#	`# mkdir /cdrom`	Indicates the root (superuser) prompt on command lines.
%	`% rlogin` remote_system_name `-l root`	Indicates the C shell prompt on command lines.
$	$ `rlogin` remote_system_name `-l root`	Indicates the Bourne shell prompt on command lines.

Window Managers and the User Interface

According to the window manager that you use, the look and feel of the user interface in CADDS can change. Refer to the following table:

Look and Feel of User Interface Elements

User Interface Element	Common Desktop Environment (CDE) on Solaris and HP	Window Manager Other Than CDE on Solaris, HP, and Windows
Option button	ON — Round, filled in the center OFF — Round, empty	ON — Diamond, filled OFF — Diamond, empty
Toggle key	ON — Square with a check mark OFF — Square, empty	ON — Square, filled OFF — Square, empty

Online User Documentation

Online documentation for each book is provided in HTML if the documentation CD-ROM is installed. You can view the online documentation in the following ways:

- From an HTML browser
- From the Information Access button on the CADDS desktop or the Local Data Manager (LDM)

Please note: The LDM is valid only for standalone CADDS.

You can also view the online documentation directly from the CD-ROM without installing it.

From an HTML Browser:

1. Navigate to the directory where the documents are installed. For example,

   `/usr/apl/cadds/data/html/htmldoc/` (UNIX)

   Drive:`\usr\apl\cadds\data\html\htmldoc\` (Windows)
2. Click `mainmenu.html`. A list of available CADDS documentation appears.
3. Click the book title you want to view.

From the Information Access Button on the CADDS Desktop or LDM:

1. Start CADDS.
2. Choose Information Access, the i button, in the top-left corner of the CADDS desktop or the LDM.
3. Choose DOCUMENTATION. A list of available CADDS documentation appears.
4. Click the book title you want to view.

From the Documentation CD-ROM:

1. Mount the documentation CD-ROM.
2. Point your browser to:

   CDROM_mount_point`/htmldoc/mainmenu.html` (UNIX)

   CDROM_Drive:`\htmldoc\mainmenu.html` (Windows)

Online Command Help

You can view the online command help directly from the CADDS desktop in the following ways:

- From the Information Access button on the CADDS desktop or the LDM
- From the command line

From the Information Access Button on the CADDS Desktop or LDM:

1. Start CADDS.
2. Choose Information Access, the i button, in the top-left corner of the CADDS desktop or the LDM.
3. Choose COMMAND HELP. The Command Help property sheet opens displaying a list of verb-noun combinations of commands.

From the Command Line: Type the exclamation mark (!) to display online documentation before typing the verb-noun combination as follows:

```
#01#!INSERT LINE
```

Printing Documentation

A PDF (Portable Document Format) file is included on the CD-ROM for each online book. See the first page of each online book for the document number referenced in the PDF file name. Check with your system administrator if you need more information.

You must have Acrobat Reader installed to view and print PDF files.

The default documentation directories are:

- `/usr/apl/cadds/data/html/pdf/`doc_number`.pdf` (UNIX)
- CDROM_Drive:`\usr\apl\cadds\data\html\pdf\`doc_number`.pdf` (Windows)

Resources and Services

For resources and services to help you with PTC (Parametric Technology Corporation) software products, see the *PTC Customer Service Guide*. It includes instructions for using the World Wide Web or fax transmissions for customer support.

Documentation Comments

PTC welcomes your suggestions and comments. You can send feedback electronically to `doc-webhelp@ptc.com`.

Chapter 1

Introduction to Concurrent Assembly Mock-Up

This chapter describes the Concurrent Assembly Mock-Up product, defines terminology, and tells you how to access its functions. The topics are:

- What You Can Do With CAMU
- Local Data Manager Menu
- Concurrent Assembly Mock-Up Windows
- The Assembly Structure
- Assembly Tree Window
- CAMU Top Bar Options
- Assembly Environment Task Sets
- Accessing Functions
- Reversing a Command
- Preparing to Work
- Accessing CAMU
- Accessing the Associative Topology Bus (ATB) Option

What You Can Do With CAMU

Using the Concurrent Assembly Mock-Up (CAMU) product you can design an overall product structure. You can use the structure to manage the overall development of the product. You can design individual part models in the context of the overall assembly, with several users working concurrently. You can define and associate information about different portions of the product structure, then extract the information using CVMAC programs.

CAMU provides a way to manage the assemblies that you design. You can see and work on both a logical representation and the geometry of the assembly.

The logical diagram shows all the assembly components and the relationships between components. While building the logical diagram, you can see the geometry of the assembly as you view and position the various models associated with the assembly structure components.

CAMU eliminates data redundancy by associating part models to components of your assembly. An assembly can have several instances of the same component, each associated with a single part. Any changes you make to that one part are reflected in all component instances associated with that part and within all assemblies that contain the same component instance.

Assemblies created using CAMU can be made up of any combination of new or existing piece parts and/or subassemblies. Piece parts or subassemblies can be designed in both the Parametric and Explicit environments - either within or outside CAMU.

Using CAMU, you can:

- Design a product structure. A logical diagram showing the relationships between different pieces of the structure. See “Building the Assembly Structure” on page 3-5.
- Have several users working concurrently on the same assembly. See “Several Users Can Work Concurrently” on page 5-27.

  Locking prevents simultaneous editing of the same portion of the assembly. See “Locking Components, Attribute Groups, and Models” on page 5-31.

  Filing adds your changes to the database. See “Filing Your Changes” on page 4-2.

  Refreshing allows all users to see the most recently filed version of the assembly structure and geometry (filed by you or another user). See “Refreshing the Main Assembly” on page 5-36.

- Include other assemblies as subassemblies in the active main assembly. See "Adding Existing Assemblies to Your Assembly" on page 3-21.
- Optionally activate and edit a reference assembly within the active main assembly. See "Changing a Reference Assembly" on page 7-25.
- Relate geometry of one part model to geometry of another, through the use of constraints. As one part is modified, other part models constrained to it can be updated. See Chapter 9, "Using Interpart Constraints".
- Use alternate orientations to isolate parts, separate subsystems, "explode" assemblies, or display parts in different locations and orientations. See Chapter 10, "Adrawings and Alternate Views".
- Manage the overall development of your product by accessing multiple parts simultaneously. You can work on one part while viewing other parts in the assembly. You can switch readily between different parts. This allows you to design each part or subassembly in the context of the assembly. See Chapter 8, "Defining Assembly Geometry".
- Define and associate non-graphic information about each piece of the structure, such as bill-of-materials information, physical properties, or labor tracking data. You can extract such information using CVMAC programs. See Chapter 11, "Defining Nongraphic Information for the Assembly".
- Create Assembly Drawing documents containing representations of geometry with HLR, dimensions, text, labels, and other annotations. See Chapter 12, "Documenting the Assembly".
- Create a new assembly database from any subassembly within the active main assembly, including a new reference assembly. You can optionally import the selected subassembly as a reference assembly within the active main assembly. See "Creating a New Assembly by Exporting" on page 3-31.

A Look at CAMU

The following sections show you the CAMU interface, including:

- The Local Data Manager (LDM) menu. It appears when you first access CAMU.
- The CAMU interface, showing a sample assembly.
- The CAMU task set icons.

See "The Assembly Structure" on page 1-7, to learn about the logical structure of an assembly and the terms used to describe it.

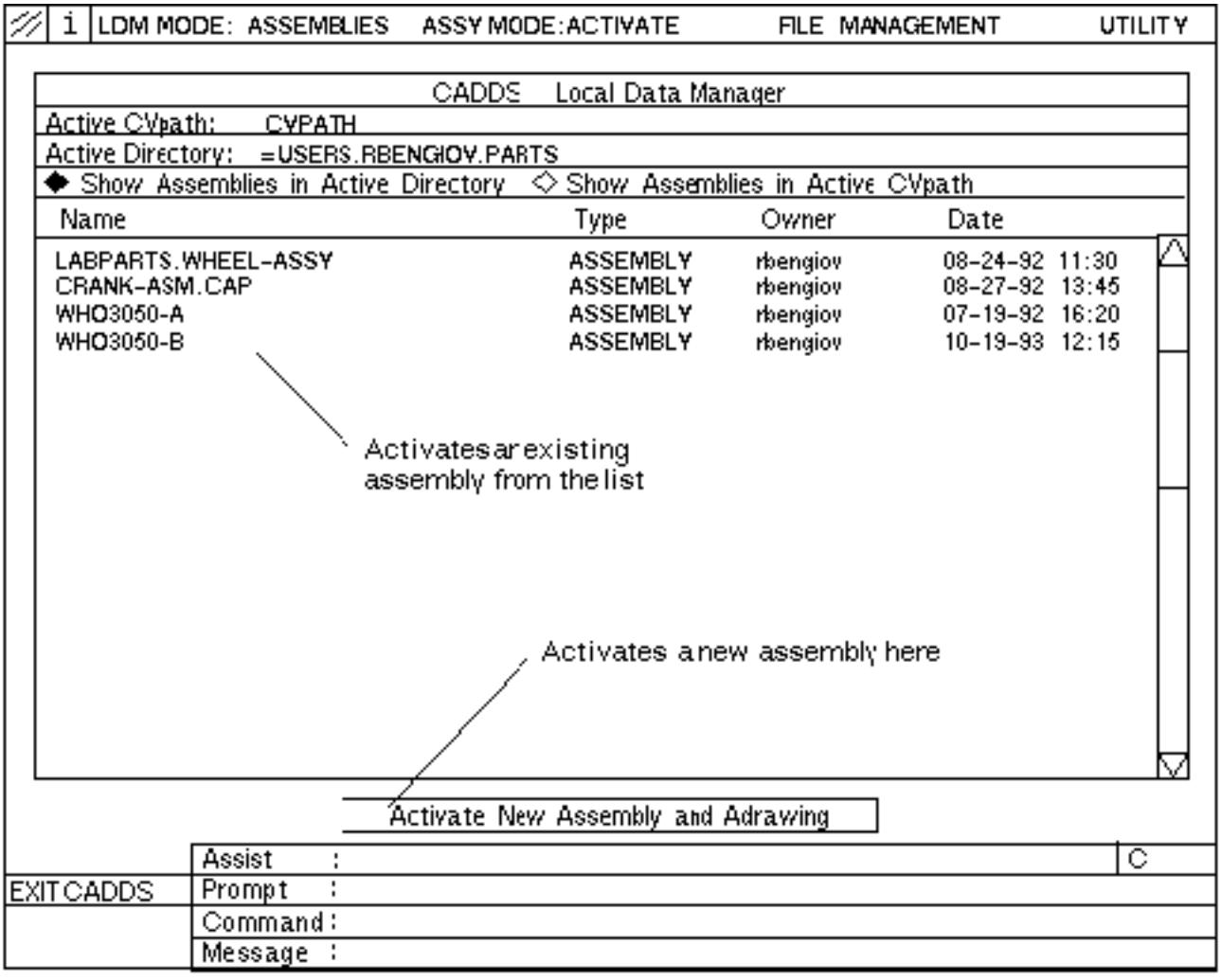
See Chapter 3, "Creating a New Assembly" to learn how to activate a new CAMU assembly.

Local Data Manager Menu

The CAMU LDM lists assemblies in your active directory or active CVPATH. You can activate a new or existing assembly from this menu.

The top bar of the CAMU LDM menu displays LDM MODE: ASSEMBLIES, indicating you are in the assembly environment. From the LDM, you can activate a new assembly (See Chapter 3, “Creating a New Assembly”) or an existing assembly (see Chapter 5, “Working in an Existing Assembly”).

Figure 1-1 Concurrent Assembly Mock-Up Local Data Manager Menu



You can toggle between Assemblies mode and Parts mode by selecting LDM MODE on the top menu bar.

Concurrent Assembly Mock-Up Windows

CAMU provides you with windows to look at the logical structure of assembly models associated with your assembly in the graphics window and to input CADDS or CAMU commands. Icons in task sets are similar to those within the CADDS Explicit and Parametric environments.

Overview of CAMU Windows

CAMU uses several windows to display specific types of information. A description and sample are given below:

- CADDS Graphics window

  In this window you build and view model of your assembly and its component subassemblies or piece parts or both.

- Explicit Text window or Parametric Command/message line

  This window displays the CADDS commands that are run using various icons and property sheets. It also contains command-specific prompts and messages, such as error messages.

  For the list of the most common errors and their solutions, refer to the section "Error Messages," in *Concurrent Assembly Mock-Up Best Practices*.

- CADDS Expanded Message Buffer

  This is the standard parametric environment option that expands the Command/Message line history.

- Assembly Structure Window

  This window shows the logical structure of the assembly. The name of the assembly, suffixed with `_1`, appears in a small box at the far left of the Assembly Structure Window. This root component represents the entire assembly. As you build your assembly, other components representing subassemblies and piece parts, are connected to each other and to this root component.

  For more information about the different components of the logical structure of your assembly, and how they relate to each other, see "The Assembly Structure" on page 1-7.

Note:

a. To display the Assembly Structure Window set the `AD_C_SERVER_ENABLED` environmental variable to `no`. The default value of this variable is `yes`.

b. To display a context-sensitive menu of available CAMU options, right-click when the pointer is in this window. See "Using the Context-Sensitive Menu Within the Assembly Structure Window" on page 1-32.

- CAMU Status Window

  This window states the name of the selected component, its associated model name, the name of the user who has locked the component, and its status (inactive, active, or suspended).

Please note: The CAMU status window is displayed along with the Assembly Structure Window only. That is, when the `AD_C_SERVER_ENABLED` environmental variable is set to `no`.

- Assembly Tree Window

  The Assembly Tree Window supplements the Assembly Structure Window. It has a tree structure with easily expandable and collapsible nodes. It is divided into two sections, the upper section containing the assembly tree and the lower section acting as the CAMU status window. See "Assembly Tree Window" on page 11.

- CAMU Status bar

  This window shows the name of the active assembly database, active part (model or Adrawing), and active modeling units.

The Assembly Structure

When you activate a new assembly, specify a name for the assembly. When the CAMU interface appears, the assembly name appears in a small box in the Assembly Structure window. This is the root component which represents the entire assembly. As you build the logical structure of the assembly, add components to represent different subassemblies or piece parts and connect them to this root component.

The Assembly Structure window shows the logical representation of your assembly.

Terminology

The following terms are used to describe the pieces that comprise the assembly structure.

Root component

This is the base of the assembly structure. It is the left most component in the Assembly Structure window representing the entire assembly. It is the highest component in the structure hierarchy and is system-generated. Each assembly can have only one root node.

You can activate the root node model within the interpart constraint process only in Parametric Assembly environment, while the same is not required or recommended in Explicit Assembly environment. The root node model is specifically designed to facilitate interpart constraints, see Chapter 9, “Using Interpart Constraints”. You must open the database prior to importing components, see “Basic Steps to Use Constraints in an Assembly” on page 9-2.

Please note: The root node facilitates the interpart constraint process only within Parametric Assembly environment, with a restriction that it does not house geometry, views of assembly components or annotations

Component

A generic term for any tree structure. A component is also referred to as a node. It represents either a piece part, subassembly, or assembly.

Component instance

A specific occurrence of a component class. The `_n` with the component name indicates it is an instance of a particular component class.

Assembly instance

A component instance representing a component of an external (referenced) assembly.

Component class

A unique name assigned to a component. Multiple instances of a component class can exist in the assembly structure. Each component class instance has the same base name, followed by an underscore and a unique system-generated number. If model information is associated with a component, all instances of that component class are associated with the same model information. Operations performed on a member of a component class are reflected in all other instances of that class, both in the structure and the graphical representation of the class.

Parent Component

A component that represents a subassembly (also called a subassembly node). A parent component must have a child component, and it can be a child of a higher level component.

Child Component

A component that is one level down from a parent component.

Siblings

Components that have the same parent. Siblings do not have to be instances of the same class.

Leaf Component

A component that has no children. Leaf components are the components typically associated with models (CADDS piece parts).

Relation Instance

The line in the assembly structure that connects one component to another, showing the parent-child relationship. In the assembly database, the relation instance is the mechanism that links the entire assembly hierarchy. These relation instances contain information such as node connectivity and orientation information between a component and its parent.

Model

A CADDS part associated with a component class within the assembly.

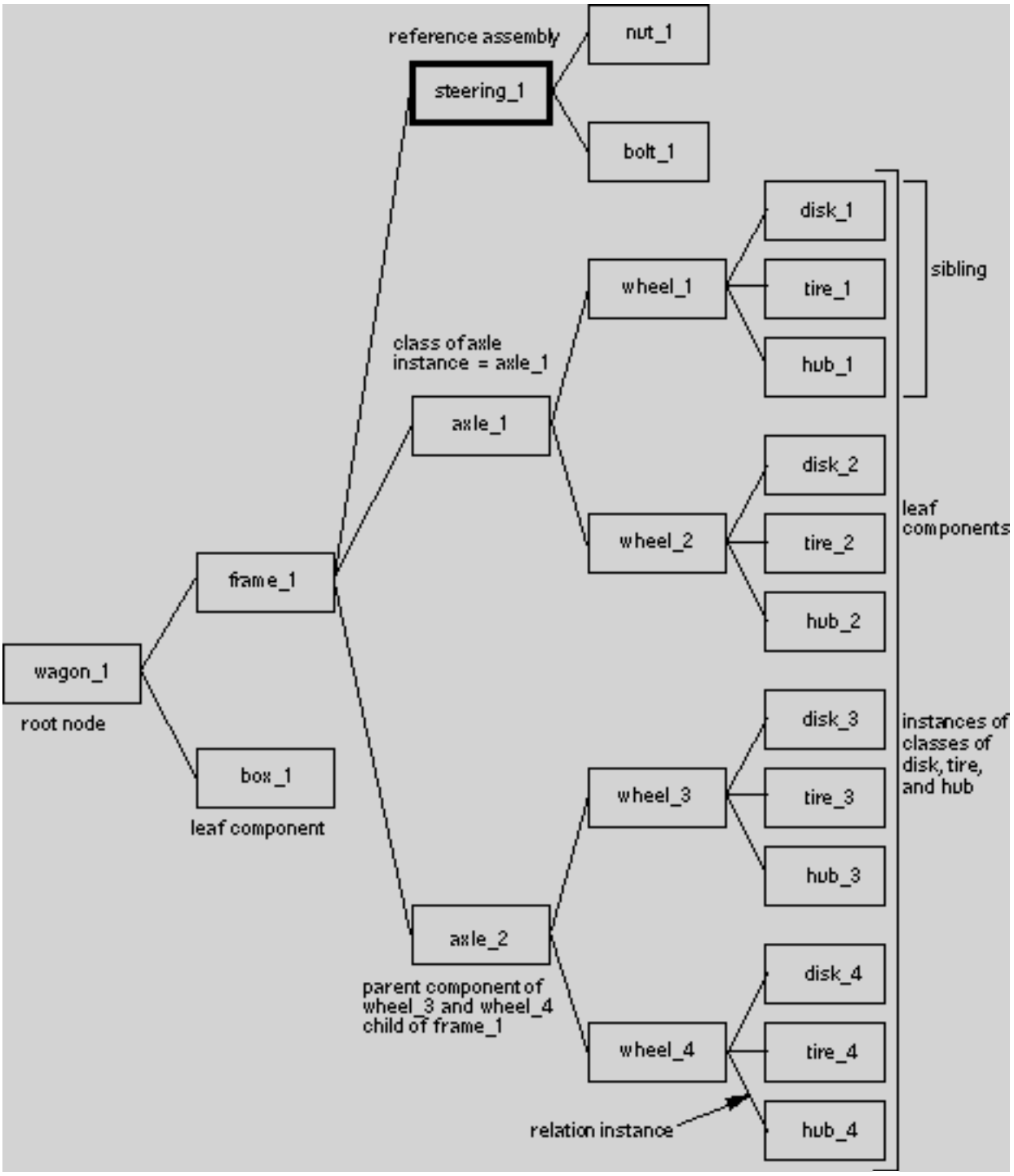
Adrawing

An Assembly Drawing. It is a part and drawing, implemented expressly for CAMU as a special CADDS part. Adrawings allow you to define views to look at the assembly, its components, and their associated models. Adrawings contain view states and layer information and are used to maintain assembly detail geometry. When you activate a new assembly, an assembly drawing file (Adrawing) called `DEFAULT` is activated. Each assembly has a minimum of one Adrawing (the `DEFAULT` Adrawing) and can have multiple Adrawings.

For more information or additional terms, see the Glossary.

The figure below shows the assembly named `wagon`, labeled with examples of assembly structure terminology.

Figure 1-2 Assembly Structure for an Assembly Named `wagon`



Please note: The assembly structure shown above is displayed in the Assembly Structure Window.

Assembly Tree Window

An Assembly Tree Window supplements the Assembly Structure window. The key features of the Assembly Tree are:

- You can easily expand or collapse any parent node that has child nodes.
- The tree display enables you to use and understand the assembly structure with ease.
- The current state of a node is represented by an icon.
- You can right-click a node for a shortcut menu with options for the node's current state.

The Assembly Tree Window is the default window. To revert to the Assembly Structure Window, set the value of the following environment variable in the `.caddsrc-local` file:

```
setenv AD_C_SERVER_ENABLED no
```

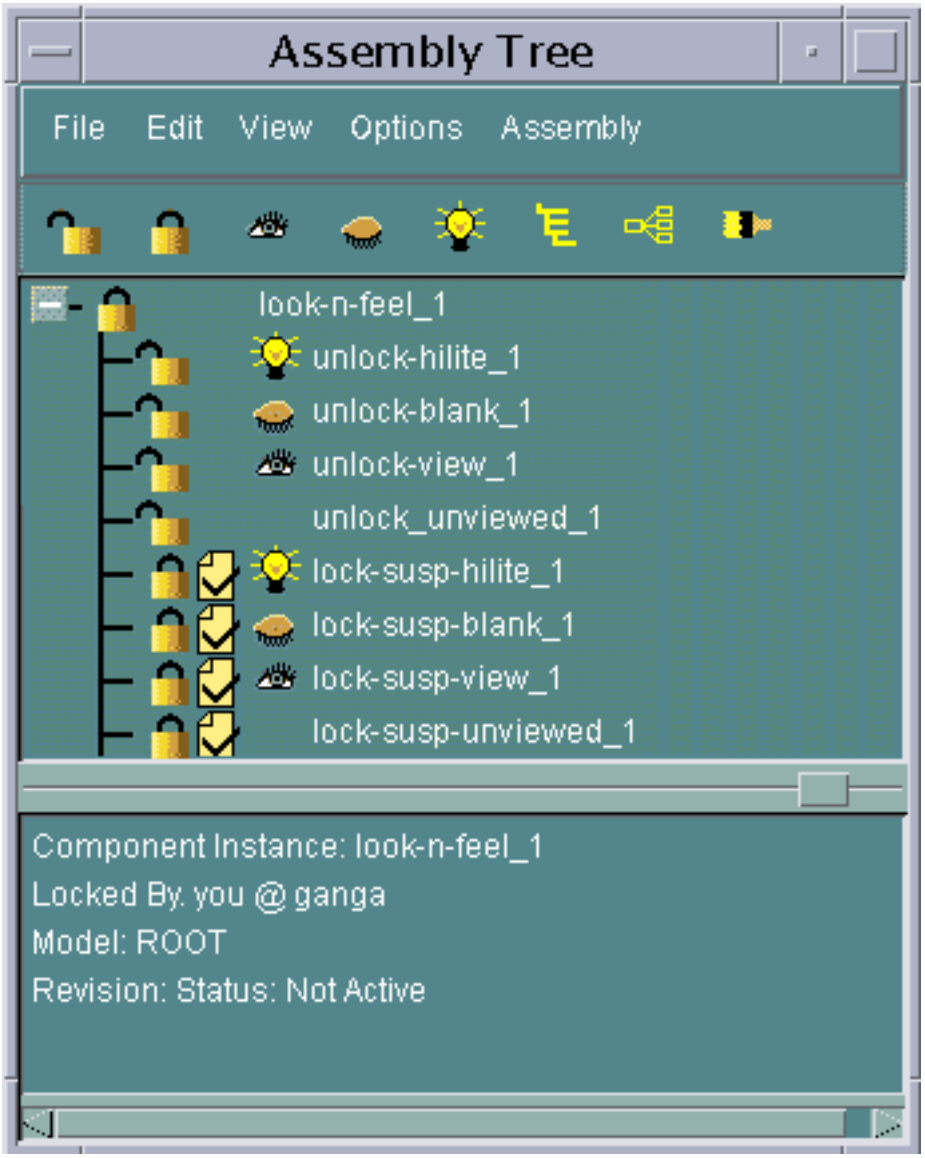
Please note: You can specify the default location and the size of the Assembly Tree Window, against the `CV_ASSEMBLY_GEOM` in the `/cadds/scripts/templates/caddsrc` file. For further information on this, please refer the `caddsrc` file.

When you activate a new or existing assembly, the assembly structure appears in a separate window. In an active assembly session, you can minimize this window but not close it.

Please note: On Windows, when you activate a new or existing assembly, the Assembly Structure opens in a separate window, which may be hidden behind the CADDS window. To bring the Assembly Tree Window to the front, click the Assembly Tree icon on the taskbar.

The following figure shows an Assembly Tree:

Figure 1-3 Assembly Tree



Please note: The CAMU Status Window is hidden when the Assembly Tree is displayed

A horizontal divider divides the Assembly Tree Window into two portions. The upper portion contains the Assembly Structure, and the lower portion, the status window. The status window displays information about the selected node in the assembly structure.

By clicking and dragging the button near the right end of the divider, you can move the divider vertically to adjust the sizes of the upper and lower portions of the window.

Please note: By moving the divider vertically, you cannot reduce the height of the assembly structure display below a minimum value.

The Assembly Structure consists of a root node of an active assembly along with its first-level child nodes. You can expand any node by right-clicking the node and choosing the Expand All option on the shortcut menu.

The types and states of the component instance nodes of the Assembly Structure are represented by different icons as shown in the following table.

Table 1-1 Node Representation in the Assembly Tree

Icon	Node Type and State
	Locked Component Node
	Unlocked Component Node
	Viewed Component Node
	Blank Component Node
	Highlighted Component Node
	Active Component Node
	Suspended Component Node
	Open Reference Node
	Closed Reference Node
	Hidden

In the Assembly Structure, the three icons that precede the component or reference assembly node denote the state of the component or reference assembly as shown in the next table.

Table 1-2 State of Component Node and Reference Assembly Node

Icon	Component Node	Reference Assembly Node
First	Locked or Unlocked	Open or Closed
Second	Active or Suspended	Locked or Unlocked
Third	Viewed, Blanked, or Highlighted	Active

A few combinations of the three icons for the state of the component or reference assembly follows.

Table 1-3 Icon Combination for Component and Reference Assembly State

Icon Combination	State of Component or Reference Assembly
	Component Locked and Highlighted
	Component Unlocked and Blank
	Component Locked, Suspended, and Viewed
	Component Locked and Active
	Reference Assembly Closed and Locked
	Reference Assembly Open, Locked, and Active

You can toggle the states of the component node by double-clicking on the corresponding icon in the assembly tree:

- Double-click the lock icon to toggle the lock status of the component.
- Double-click the eye icon to toggle between the view and unview states of the component.
  - Double-click the highlight icon to unhighlight the component.
  - Double-click the blank icon to unblank the component.
- Double click the folder icon to toggle between the closed and open states of the assembly node.

Shortcut Menus within the Assembly Tree Window

You can access CAMU options from the Assembly Tree Window. To access these options, right-click within the Assembly Tree Window. A dynamic, shortcut menu appears. For example, if you select a component node and right-click, the shortcut menu lists the options for use on the selected component node.

The following table lists the options on the shortcut menu, which depend upon the type of node you select in the Assembly Tree Window:

Table 1-4 Options on the Assembly Tree Window Shortcut Menus

Node Selected	Option	Suboption
None	New Adrawing	
	Old Adrawing	
	Save Viewstates	
	File Adraws/ Models	
	List Adrawings	
	List Assemblies	
	Refresh Assembly	
	Scroll Structure	
	Delete Adrawing	
	Undo Acommand	

Table 1-4 Options on the Assembly Tree Window Shortcut Menus

Node Selected	Option	Suboption
Component node	Activate	
	Add	Assembly
		Component
	Blank	Blank Component
		Blank Subassembly
		Unblank Subassembly
		Advanced
	Change	
	Copy	Component
		Mirror
	Delete	Component
		All Instance
	Expand All or Collapse All	
	Export	
	Highlight	Highlight Component
		Highlight Subassembly
		Unhighlight Subassembly
		Advanced
	Lock	Lock Component
		Lock Subassembly
		Unlock Subassembly
		Advanced
	Move	
	Quit	
	View	View Component
		View Subassembly
		Advanced
	Render	ShadeNoWire
		WireFrame

Table 1-4 Options on the Assembly Tree Window Shortcut Menus

Node Selected	Option	Suboption
Component node	Reorder	
	Report	Brief
		Detailed
	Translucent	

Table 1-4 Options on the Assembly Tree Window Shortcut Menus

Node Selected	Option	Suboption
Reference Assembly node	Activate	
	Add	Component
		Assembly
	Blank	Blank Component
		Blank Subassembly
		Unblank Subassembly
		Advanced
	Change	
	Copy	Reference
		Mirror
	Delete	Reference
		All Instance
	Expand All or Collapse All	
	Export	
	Highlight	Highlight Component
		Highlight Subassembly
		Unhighlight Subassembly
		Advanced
	Lock	Lock Component
		Lock Subassembly
		Unlock Subassembly
		Advanced
	Move	
	Open	Next Level
		All
		Advanced
	Close	
	Quit	Reference
		Advanced

Table 1-4 Options on the Assembly Tree Window Shortcut Menus

Node Selected	Option	Suboption
Reference Assembly node	View	View Component
		View Subassembly
		Advanced
	Reorder	
	Report	Brief
		Detailed

The Advanced option appears on the shortcut menus. Choosing Advanced displays the property sheet for the selected option. You can then specify more details for the operation. For example, if you select the View option from the shortcut menu, the component appears without any attributes. Choose the Advanced option to specify other attributes such as parameter, layer, and so on.

The contents of the shortcut menu depend upon the state of the selected component node. For example, if you select a locked node, the Unlock option appears on the shortcut menu. If the node is unlocked, the Lock option appears. The following table lists options for each state of a selected component.

Table 1-5 Options Displayed on the Shortcut Menus

State of the Node	Option
Blanked	Unblank
Unblanked	Blank
Highlighted	Unhighlight
Unhighlighted	Highlight
Locked	Unlock
Unlocked	Lock
Viewed	Unview
Unviewed	View

Assembly Tree Window Top Bar Options

You can perform some operations on the whole assembly rather than on an individual component. The following options are available from the top bar menu on the Assembly Tree Window.

- File - Provides options for filing the assembly, printing the assembly structure, setting the page size and orientation, and for exiting the assembly session.

- Edit - Provides options for adding, changing, copying, deleting, and for exporting a component.
- View - Provides options for blanking or unblanking, highlighting or removing the highlight, displaying or unviewing a selected component, displaying or hiding the tooltip text and for displaying either the part names or the model names of the components. Also provides options for expanding all nodes of the collapsed assembly structure, generating a report, and for generating a list of Adrawings or assemblies.
- Options - Provides options for specifying font size, font style, and the look and feel of the Assembly Tree.
- Assembly - This option appears only when you select a component on the Assembly Tree. It provides options for activating, locking, or moving the component, and opening the reference assembly.

Assembly Tree Window Toolbar Options

You can perform some operations on the selected tree node in the assembly tree using the toolbar options. The Assembly Tree Window toolbar contains the following options. These options may be classified into two types.

Unlock:

Unlocks the selected component node.

Lock:

Locks the selected component node.

Unblank:

Unblanks the seleced component node.

Blank:

Blanks the selected component node.

Highlight:

Highlights the selected component node.

Display Model Names or Component Names:

This is a two-state toggle button which has two different forms as follows:

Displays the Component names of all the component nodes.

Displays the Model names for all the component nodes.

This option modifies only the display of the node text in the Assembly Tree Window. It does not perform any action on the actual assembly.

Issue Commands for Node or the Entire Sub-assembly:

This is a two-state toggle button which has two different forms as follows:

Specifies that the lock, unlock, blank, unblank and highlight operations performed further should affect only the selected component node.

Specifies that the lock, unlock, blank, unblank and highlight operations performed further should affect the selected component node and all its sub nodes.

Refresh:

Click this option to refresh the Assembly Tree Window whenever required.

Please note: The Refresh option refreshes only the display of the assembly tree in the Assembly Tree Window. This option is different from the REFRESH ASSEMBLY command.

CAMU Top Bar Options

The options available from the CAMU Top Bar are the same as those in the Explicit and Parametric modeling environments. For information about these options, see *Introduction to CADDS 5*.

The following options of the top menu bar are specific to CAMU:

- FILE - Provides options for filing, exiting, and quitting the assembly structure, Adrawing(s), and/or model(s), and for exiting CADDS.
- ADRAWING - Provides commands specific to Adrawings, including Activate New, Activate Old, Change, Delete, Toggle Exploded View, and Delete Exploded View. Selections are available for plotting and for changing the background color. You have choices for Copy and Enlarge in the Explicit environment.
- COMP/ASSY - Provides graphics display options for components, including Change Layer, Highlight Component, View Component, Blank Component, Unblank Component, Scroll, Zoom, and Highlight Active.The explicit environment has an option for Selection Mode.
- The Query icon (question mark) launches a menu with these CAMU selections:
  - Verify Object - Shows who has locked a component, the model association, the view status, and orientation of the groups attached to the component.
  - Print Component - Shows, in the report window, an indented listing of the assembly structure hierarchy downward from the component that you select.
  - List Adrawings - Shows a list of all Adrawings associated with the assembly, and their status.

Assembly Environment Task Sets

Both Assemblies in the Parametric and Explicit environments provide similar functions. In addition, from the Assembly in Parametric environment you can access the CONSTRAINTS task set, as well as the Explicit and Parametric Modeling environments. From the Assembly in Explicit environment, you can access the ATTRIBUTES task set and the Explicit Modeling, Parametric Modeling, and Manufacture environments. The Modeling pulldown menu allows you to access other environments from within CAMU.

This section shows the tasks you can access from the global menu in both environments. It also shows the environments you can access from the Modeling pulldown menu.

Common Tasks

You can access labeled functions from both the ASSEMBLY/EXP and ASSEMBLY/PAR global menus. The upper portion includes icons for Refresh Assembly, Undo Acommand, Lock/unlock Component/model, Move Component (translate, rotate, orient, and position), Activate Model, and Copy Component.

The portion of the menu under COMP/ASSY (the taskset) includes Add Component, Change Component, Remove Component and Assemblies, Compressing, Blanking, Reordering, Mirroring Components in the structure.

Add Component And Add Assembly are discussed in Chapter 3, “Creating a New Assembly”. The other functions are discussed in Chapter 7, “Changing the Assembly Structure”.

Figure 1-4 Assembly Taskset for Parametric and Explicit Assembly Environments

Moving to Other Environments and Task Sets

You can move to other environments and task sets without having to file and exit your active main assembly. Both the ASSEMBLY/PAR and ASSEMBLY/EXP global menus provide options for moving to the Explicit and Parametric modeling environments, as well as other environments.

When you move to different environments, for example, from Explicit to Parametric modeling, the view state within the Graphics window is maintained.

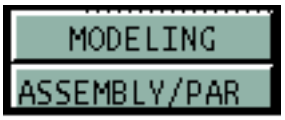
The ASSEMBLY/PAR, ASSEMBLY/EXP, and COMP/ASSY menus and task sets are illustrated on the following pages.

Please note: If a menu option is hazed, you cannot select it.

You can move to another environment task set, such as Parametric Modeling, from CAMU.

1. Select ASSEMBLY/PAR (or whatever environment is active) from the top left corner of the active task set.

Figure 1-5 Moving to the Other Environment Groups



2. Select the desired environment task set from the resultant menu

Figure 1-6 Moving to Other Environments and Task Sets

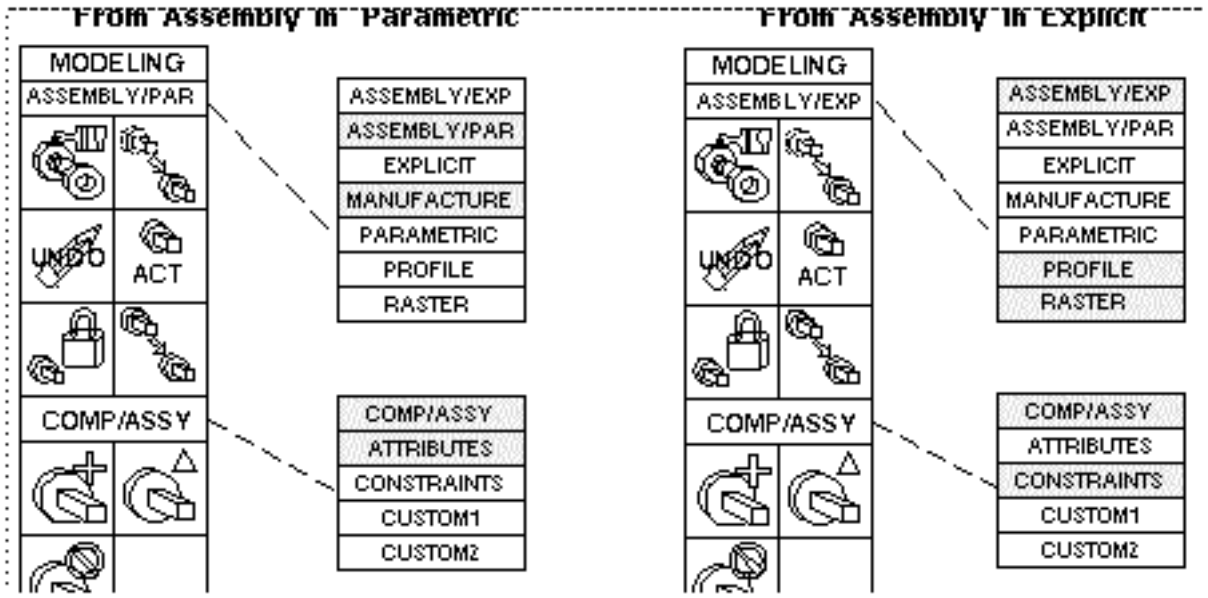


Figure 1-7 Accessing the Constraints Task Set from Assembly in Parametric

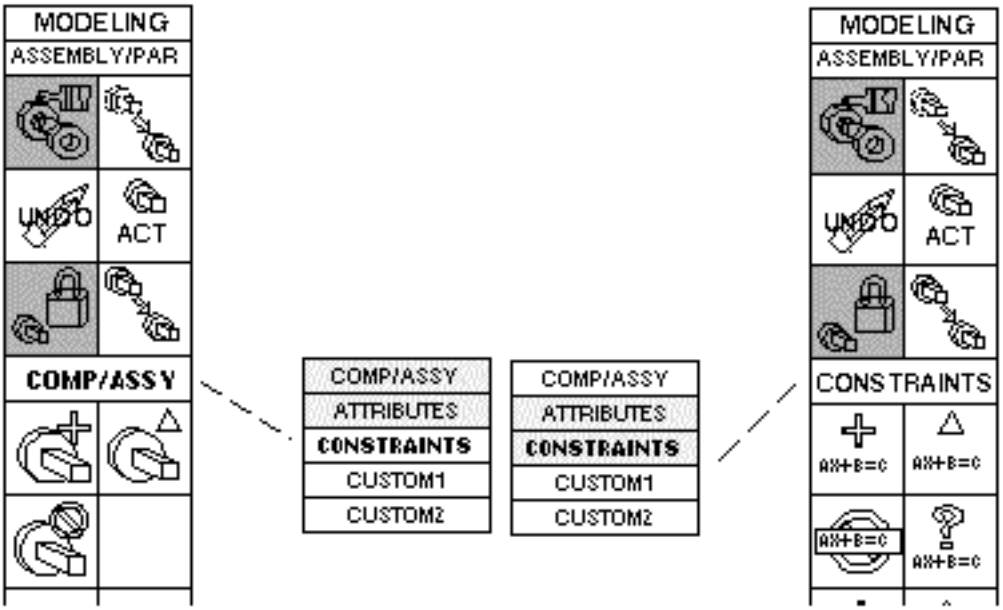
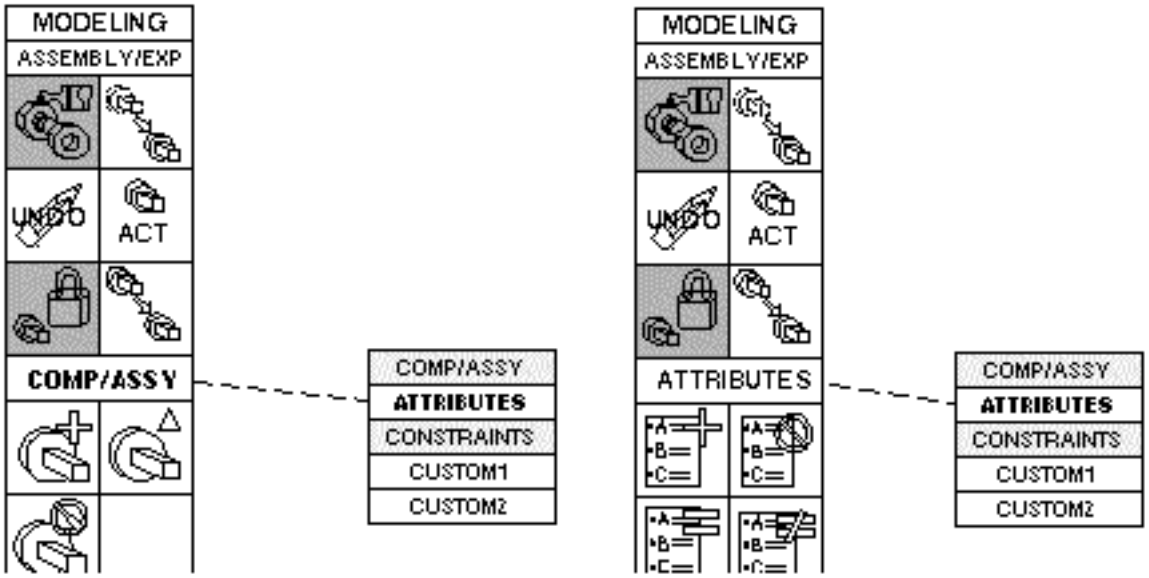


Figure 1-8 Accessing the Attributes Task Set from Assembly in Explicit

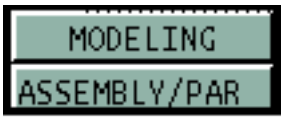


Moving to Other Environment Groups

You can move to an application task set within another environment group, such as Harness Layout, from within CAMU.

1. Select MODELING (or whatever group is active) from the top left corner of the active task set.

Figure 1-9 Moving to the Other Environment Groups



2. Select the desired application task set from the resultant menu.

Accessing Functions

The methods used to select options and fill in property sheets are the same in both the Explicit and Parametric modeling environments, with the addition of features particular to CAMU. This section provides brief descriptions of how to use icons and windows.

Using an Icon to Select an Option

Move the cursor on the icon. Click the left button. One of the following occurs:

- The command is processed. In the Explicit environment, you can read the status of the command in the Text window. When processing is done, the CADDS prompt reappears. The command is executed in the Parametric environment.
- A menu of additional commands appears. Choose the function you want. Click the left mouse button.
- To give additional information, fill in the property sheet that is displayed.

Property Sheets

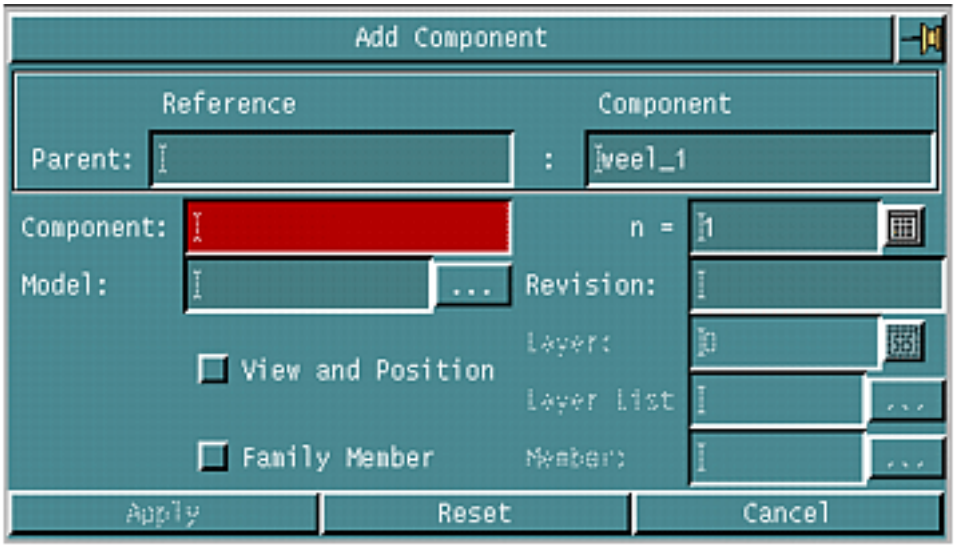
Property sheets have fields in which you provide information. A red vertical line indicates a field in which you enter information. The methods used to fill in property sheets are similar to those used in the Explicit and Parametric modeling environments.

With CAMU you can select a component from the Assembly Structure. CAMU property sheets provide input options. Rectangular fields on property sheets provide a place for you to fill in other information.

Filling In Fields on a Property Sheet

To move the cursor to a different field on the property sheet, either press RETURN key, or select the desired box with the help of left mouse button. Fill information in a field on a property sheet by typing it in. You can also do one of the following, depending on the type of information appropriate for the field:

Figure 1-10 Example of a Property Sheet



- Choose a component from the structure.

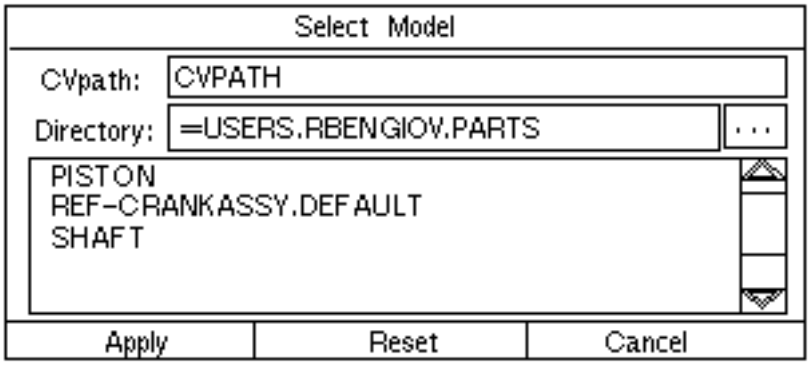
  Some property sheets require two component entries.

  To fill in the Component field, select the component from the assembly structure.

  To fill in the Parent field, use the middle mouse button to select the parent component from the assembly structure.

- Choose an item from a property sheet that appears.

Some property sheets display a three dot options button. For example, in the Add Component property sheet, clicking the three dot options button by the side of Model field, displays a run time list of existing parts in your CVPATH.



Choose the model from the list. It gets highlighted. The highlighted model name appears in the field on the property sheet If you do not want to generate the list, type in your entry.

- Select a component from the graphics display of your assembly structure.

  The component name that is linked to this model instance appears in the field on the property sheet. Each property sheet has one field that can be filled in this way. This is similar to selecting from the structure.

- Use the calculator to provide a number.

  Click the calculator to enter numbers in a field. Use numbers and functions as on a regular calculator. After selecting the numbers, click the answer. The number appears in the appropriate field on the property sheet.

Using Pushbuttons on Property Sheets

Pushbuttons on property sheets are for selecting an option. When you select one, it is red. Pushbuttons are of two types:

- Square button (2-way toggle) — Toggles a selection on or off.
- Diamond-shaped button — Allows you to select only one of several choices listed. When one button is selected from a group, other buttons in the group are automatically deselected.

Processing the Property Sheet

After providing the necessary information on the property sheet, process it by selecting one of these options from its bottom bar:

- Apply — Processes the command or function.
- Reset — Returns all information on the property sheet to its default state.
- Cancel — Closes the property sheet and does not process the command.

Pushpin Icon

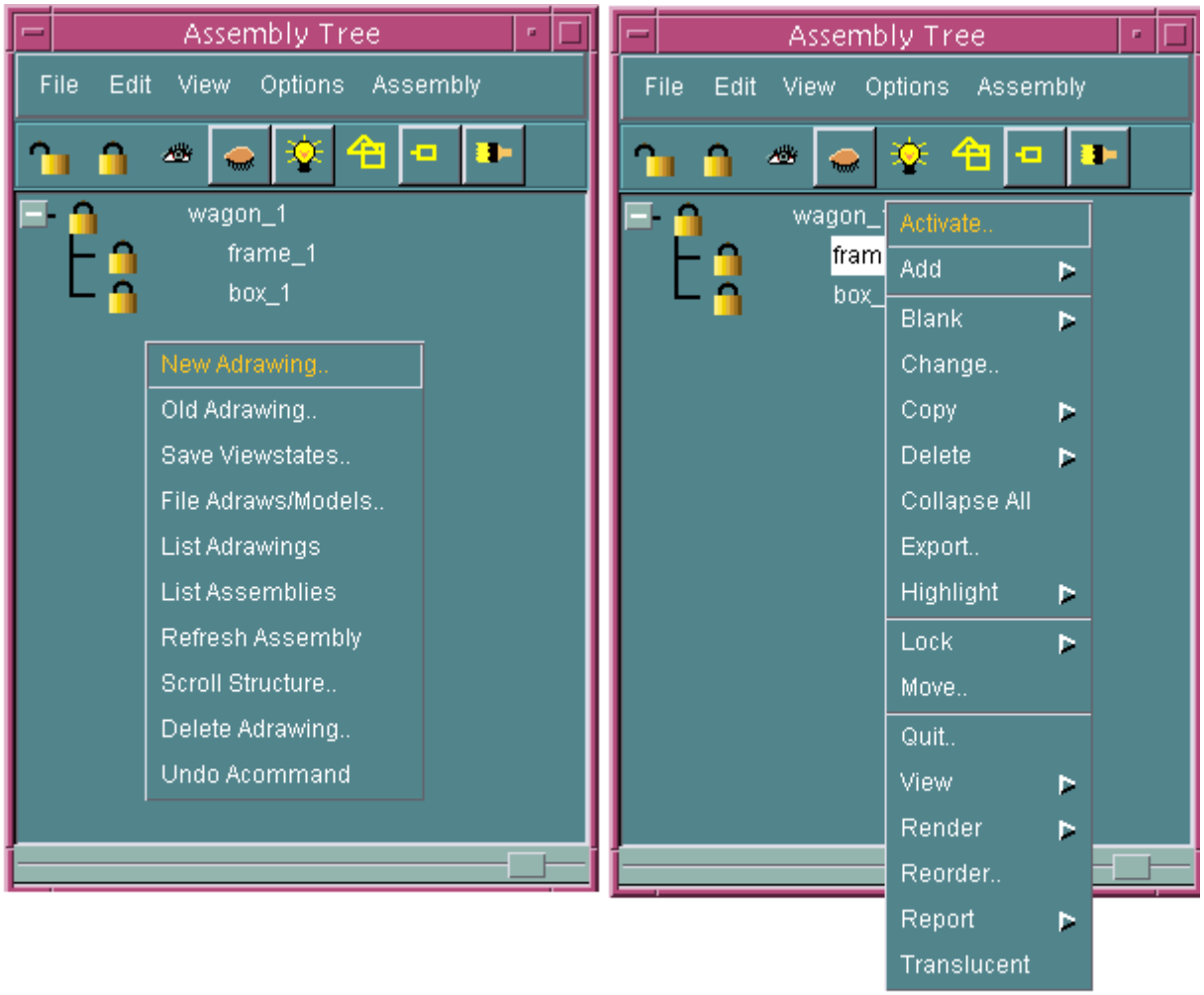
Select the pushpin at the top right of the property sheet to keep the property sheet on the screen even after you choose Apply or Reset option. Click Cancel to cancel the pushpin.

Using the Context-Sensitive Menu Within the Assembly Structure Window

You can access CAMU commands from the Assembly Structure window. With the cursor inside the window, click and hold the right mouse button.You get a context-sensitive menu. It shows all the commands that are immediately available to you. When you select a component, its box is shaded. If a property sheet appears, you can see the component name in one of the fields on the property sheet.

The figure below shows two examples of this context-sensitive menu. A star (★) in the small box indicates the component is locked, giving you write access to it. Locking is explained further in Chapter 4, Working in an Existing Assembly.

Figure 1-11 Example of a Context-Sensitive Menu in the Assembly Structure Window



Menu displays commands available when no component is selected.

Menu displays commands available when a component is selected.

Please note: Export Assembly is also available when a component is selected.

Using Online Documentation

This user manual is supplied in HTML and PostScript form - see Preface of this User Guide. Select the Information Access icon on the top bar menu. Select DOCUMENTATION. The system displays a list of available books.

Commands and Command Help

You can use icons to perform functions in CAMU or you can enter commands in the Text window. Some icons perform the functions of one or more commands. To list the commands incorporated in a particular icon, move the cursor over the icon, and press the HELP key. The resultant screen shows a menu listing all commands represented by the icon.

Use online command help to get information on the desired command.

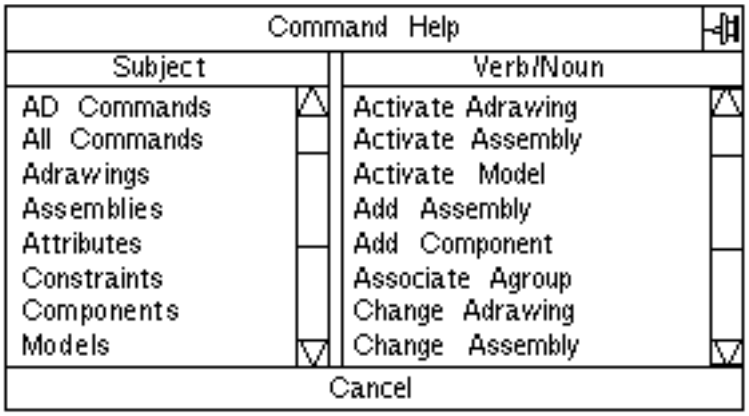
Assist Lines

The assist line provides brief, on-screen help. It appears by default unless specified otherwise in the `.caddsrc` or `.caddsrc-local` file. If not, select Assistance from the Information Access icon (i) at the upper left of the screen. An assist line appears in the bottom portion of the screen. As you move the cursor over each icon, a description of the icon's function appears in the assist line.

Using Online Command Help

Online information for a specific function is available from the Information Access icon on the top bar menu. Select COMMAND HELP. The system displays a property sheet containing two lists:

- Subject
- Verb/Noun



Based on your choice from the Subject list, the Verb/Noun list displays the commands related to the chosen subject area. The default Subject—Verb/Noun list display is All Commands.

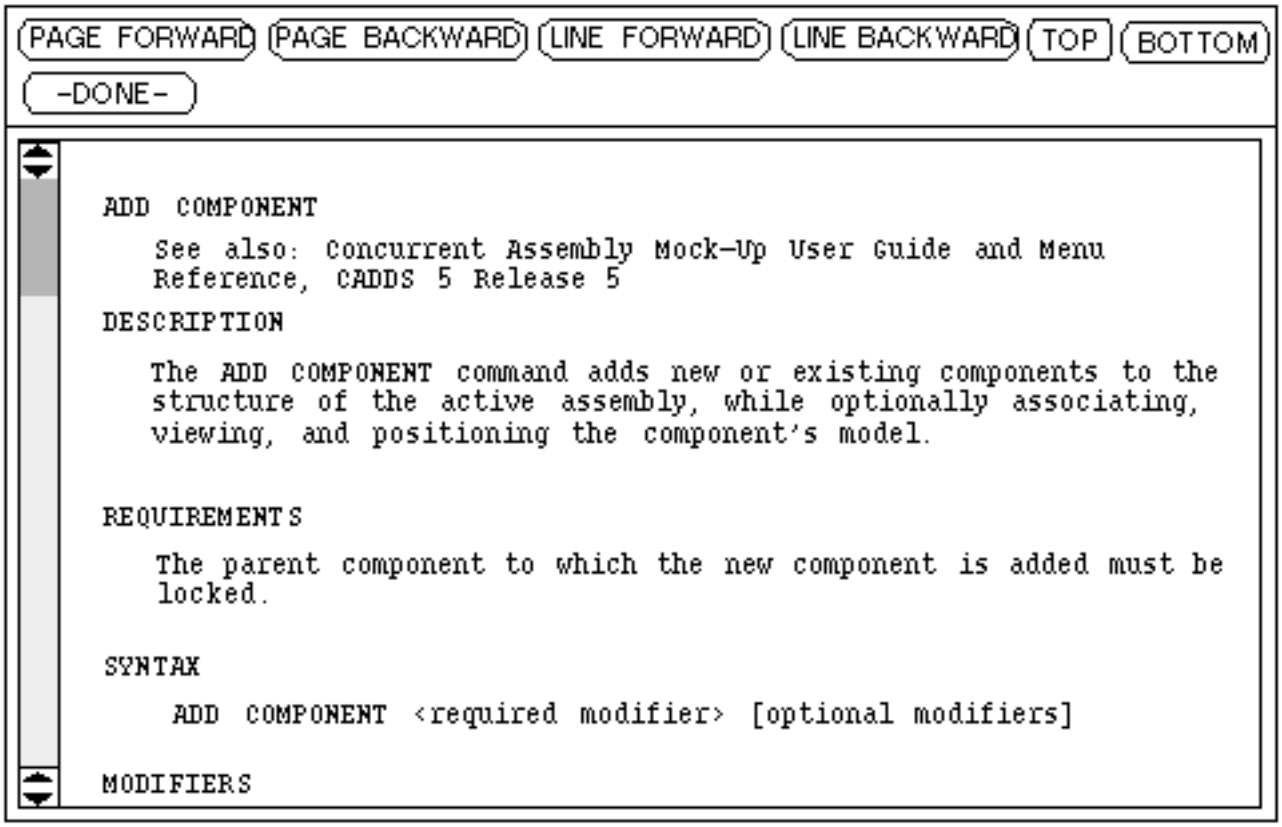
To display online command information for a specific task or function:

1. Select the Information Access icon from the top bar menu and the Command Help option from the menu.
2. Choose the subject area from the displayed list, for example, Components.
3. Choose the verb/noun combination that best describes the task/function, for example, ADD COMPONENT.

The system responds with a window containing information that describes the verb/noun selected. The information includes:

- General description
- The valid mode (Model or Draw), in which the task/function operates
- Syntax
- Descriptions of modifiers used with the Verb/Noun command

Figure 1-12 CADDS Viewfile Window: Add/Component


PAGE FORWARD
PAGE BACKWARD
LINE FORWARD
LINE BACKWARD
TOP
BOTTOM
-DONE-
ADD COMPONENT
See also: Concurrent Assembly Mock-Up User Guide and Menu Reference, CADDS 5 Release 5
DESCRIPTION
The ADD COMPONENT command adds new or existing components to the structure of the active assembly, while optionally associating, viewing, and positioning the component's model.
REQUIREMENTS
The parent component to which the new component is added must be locked.
SYNTAX
ADD COMPONENT <required modifier> [optional modifiers]
MODIFIERS

You can browse through the text by:

- Selecting browsing options from the top bar of the window
- Manipulating the scroll bar along the left side of the window

Click -Done- from the top bar to close the window.

Reversing a Command

To reverse the last CAMU command or function you have executed, click Undo from the ASSEMBLY/PAR menu task set, located at the upper left corner of the screen. The assembly returns to its former status.

Please note: Undo is also available on the context-sensitive pulldown menu.

When you click Undo, the latest command you are able to undo is undone. If you click Undo again, the previous command is undone. You can continue this until you reach a command that cannot be undone.

Commands You Cannot Undo

Any command issued before filing the assembly cannot be undone.

The following commands cannot be undone:

- ACTIVATE ASSEMBLY
- ACTIVATE MODEL
- ACTIVATE ADRAWING
- LIST ADRAW
- VERIFY OBJECT
- REFRESH ASSEMBLY
- MODIFY APPEARANCE
- LOCK COMPONENT
- UNLOCK COMPONENT
- SELECT ASSEMBLE

In addition, none of the inter-part constraint commands can be undone except Change Component Name (on an imported component). For more information about inter-part constraints, see Chapter 9, “Using Interpart Constraints”.

Preparing to Work

Before you begin working in CAMU, be sure that the software is installed and your system is properly configured.

Viewing Parts

To ensure that you and other users can view the parts that you create and/or use in CAMU, the following entry in the `.caddsrc-local` file is set to 'yes' so that a `vp_links` directory is automatically created for each new part.

```
setenv CADDSENABVP 'yes'
```

Entering CAMU

Enter `cadds5` in a command shell window. This automatically brings up the CAMU LDM menu, if the following environment variable has been set in your `.caddsrc-local` file. If this variable is not set, type `cadds5 -assembly` to enter CAMU.

```
CV_UI_DEFAULT_ENV 'assembly'
```

Manipulating a Component That Has No Geometry

The `CV_ASSEMBLY_HANDLE_PART` environment variable establishes the name of the handlepart that you must create to position a component that has no associated model geometry. Set the default environment variable as follows.

```
setenv CV_ASSEMBLY_HANDLE_PART 'adhandlepart'
```

Any existing `ahandlepart` should contain the updated `_gr` file. To ensure that the `_gr` file is up-to-date, do one of the following steps:

- Activate the part in the Explicit environment. Run the Render View or Change Renderview command with any of the shade or HLR rendering options. File the part.
- Run the `update_gr` script.

Create a part to serve as the handlepart. The handlepart must have at least one entity. Do one of the following:

- Create a part that serves as the handlepart and specify the part's name in the `CV_ASSEMBLY_HANDLE_PART` variable. Access this part via the CVPATH or by using a full CADDS style path name.

- Activate a part in your `Create` directory to serve as your handlepart. Follow the guidelines in the following file to create the handlepart:

  `/usr/apl/cadds/data/am/supplements/README_1`

Establishing the DB Daemon

In a single-user CAMU environment, the DB daemon (ODB_DAEMON) runs locally on your machine. It is started when you invoke CAMU or when you run the `db_daemon` script. In a multi-user CAMU environment, the DB daemon is used to control a multi-user CAMU session (using a spawned ODB_SERVER process tied to the assembly upon its activation) and therefore must be established on a host machine. Each user who accesses a particular assembly must access the same DB daemon, and specifically the same ODB_SERVER process. The following line establishes the host system on which the DB daemon is to reside. $CADDSHOST can be a local or remote system name:

```
setenv DB_DAEMON_HOST $CADDSHOST
```

For more information, see Chapter 2, “Configuring Concurrent Assembly Mock-Up”.

Suspending a Specified Number of Models

To suspend a specific number of models to solve/regenerate the components to which they are associated, set the variable with the maximum number of models you want to be able to suspend (plus 1 for the active model). The default is 6. See the section “Examples of Moving Between Adrawings and Models” in Chapter 5 for more information about suspending models.

```
setenv CADDS_MAX_PART nn
```

Where `nn` is the number of parts (models and/or Adrawings) you want to be able to suspend.

Please note: For additional information see the template files in `/usr/apl/cadds/scripts/templates/.caddsrc`

Accessing CAMU

Access CAMU from the window environment or from the CADDS LDM menu.

To access CAMU from your system's window environment, do the following:

1. Log in to your system and start your window environment.
2. After the window environment appears, open a shell window. At the prompt, type `cadds5 -assembly`, or `cadds5` if CV_UI_DEFAULT_ENV is set to -assembly. Press `RETURN`. The CAMU LDM menu appears.

If you have been working in CADDS, you can access CAMU from the CADDS LDM menu, as follows:

1. On the top bar of the CADDS LDM menu, click LDM Mode with the left mouse button.
2. From the choices, select Assemblies. The LDM for CAMU appears.

The CAMU LDM menu displays the active CVPATH, the active directory, and a list of the assemblies in your current working directory. You can also display a list of parts in your active CVPATH.

From the LDM menu you can activate a new or existing assembly, specifying the environment.

Accessing the Associative Topology Bus (ATB) Option

You can use the ATB option to share the part and assembly data across CADDS 5 and Pro/ENGINEER. You can access Pro/ENGINEER parts or assemblies in the Assembly environment. You can also share part geometry, topology, and assembly data in CADDS 5 and Pro/ENGINEER in an associative manner.

You can export a part or assembly from CADDS 5 or Pro/ENGINEER as a Translated Image Model (TIM) part or assembly and also import these TIM parts or assemblies in CADDS 5 or Pro/ENGINEER. A TIM part is a part that is translated from a source part originally created in a different CAD system.

For example, you can export a CADDS 5 assembly as Pro/ENGINEER TIM assembly and activate this Pro/ENGINEER TIM assembly in Pro/ENGINEER. Similarly, you can export a Pro/ENGINEER assembly as a CADDS 5 TIM assembly and also activate this CADDS 5 TIM assembly in CADDS 5.

Please note: Use CADDS 5 precision parts in ATB Enabled CADDS 5. ATB Enabled CADDS 5 does not work on CADDS 4X single and double precision parts.

You can access the ATB option either in the LDM mode or in the Assembly mode.

The LDM screen lists parts or assemblies, either in your active directory or the active CVPATH, depending on the mode of selection. The LDM also displays a list of CADDS TIM parts and assemblies in your active directory.

You can perform the following ATB operations:

- Import a Pro/ENGINEER part or assembly as a CADDS 5 TIM part or assembly.
- Export a CADDS 5 part or assembly as a Pro/ENGINEER TIM part or assembly.
- Verify a CADDS 5 TIM or Pro/ENGINEER TIM part or assembly.
- Update a CADDS 5 TIM or Pro/ENGINEER TIM part or assembly.

You can perform these operations on single or multiple parts and assemblies. You can also these operations in the NAV-CAMU mode.

A single user or different users can perform these multiple operations simultaneously.

Refer to *Using Associative Topology Bus Enabled CADDS 5i* for more details on using the ATB feature.

Please note: Before starting an ATB session, set the `CADDS_ATB_ENABLED` environment variable in your `.caddsrc-local` file as follows:

```
setenv CADDS_ATB_ENABLED yes
```

You can also start an ATB session by using the `-atb` command line option:

```
cadds5 -atb
```

Updating TIM Parts with Pending Regeneration State

If you have added geometry or features to a CADDS TIM part, you must update the part and regenerate the part on subsequent activation. Such TIM parts are called parts in pending regeneration state.

To complete the associative update process of a TIM part in pending regeneration state, perform the following steps:

1. Activate the CADDS TIM part in the Parametric environment.

Please note: You cannot activate TIM parts that are in pending regeneration state in the Explicit environment.

2. Activate a drawing.

   The Parametric geometry that is dependent on the imported TIM part geometry, is regenerated automatically.

Please note: If regeneration fails, the failed command and all subsequent commands that have not been regenerated, are removed.

3. File the part.

   The changes are permanent.

Querying the ATB Transactions

In an ATB session, each of the Import, Export, Verify, Update, and the show Pro/ENGINNER part and Pro/ENGINEER TIM part or assembly operations is uniquely identified by a transaction number starting from zero. The Report Window displays the beginning and end of each transaction. In addition, an alert message appears at the end of the transaction or in case of an error.

You can query for the status of the transaction table in CADDS 5 and print the status to the Report Window. In the Parametric environment, you can also print the status to the log file.

The Topology bus report lists details of each transaction.

In the Parametric environment, you can log the contents of the Report Window in a report file. This report file is stored in the `_bcd` directory as an external ASCII file.

You can control the display of the alert messages by setting the `ATB_ALERT_POLICY` variable appropriately in the `.caddsrc-local` file. Set the `ATB_ALERT_POLICY` variable as follows to display an alert message after each successful or failed transaction:

```
setenv ATB_ALERT_POLICY ALWAYS
```

Set the `ATB_ALERT_POLICY` variable as follows it to display the alert message only after a failed transaction:

```
setenv ATB_ALERT_POLICY ONERROR
```

Check the Report Window for the details of each transaction.

Use the Generate Topology Bus Report option from the Query option on the top bar of the CADDS window to obtain the following types of reports:

- Specific transactions
- A range of transactions
- All transactions
- Successes or failures of each part for an imported or exported assembly transaction

Chapter 2

Configuring Concurrent Assembly Mock-Up

This chapter provides the information you need to configure Concurrent Assembly Mock-Up so that several users can concurrently access the assembly database. It also provides information about setting up environment variables within your `.caddsrc-local` file. The topics are:

- Implementing CAMU and EPD
- .caddsrc-local File Information
- Configuring Multi-User CAMU
- Arranging Projects Using CVPATH
- Data Management

Implementing CAMU and EPD

CAMU is an integral part of Electronic Product Definition (EPD). Deciding to implement CAMU and EPD is a strategic decision that may have a profound and far reaching effect on the way your enterprise operates. Although EPD focuses on the product development process, it's impact is much wider. Successfully implementing CAMU and EPD involves more than technology. It also involves processes, people, and their interaction.

You can engage our industry consultants to perform a Product Development Diagnostic (PDD) to provide a business context and framework for implementing EPD within your organization or enterprise.

.caddsrc-local File Information

To run CAMU, set the following variables in the default `.caddsrc` file, `/usr/apl/cadds/scripts/templates/.caddsrc`. The default `.caddsrc` file contains all the required variables for running CAMU.

Please note: To customize your environment, create a `.caddsrc-local` file in your home directory and add these variables. You can copy the template version into your home directory as `.caddsrc-local` and modify it to contain only the custom settings.

- DB_DAEMON_HOST specifies the target system on which the CAMU database server (ODB_SERVER) runs. For multi-user configurations, each user needs to connect to a common database server. By default, ODB_SERVER runs on your local machine (machine_name).

  ```
  setenv DB_DAEMON_HOST 'machine_name'
  ```

- ODB_am_wh sets the name of ODB database server to ODB_SERVER.

  ```
  setenv ODB_am_wh /usr/apl/cadds/bin/ODB_SERVER
  ```

- CV_ASSEMBLY_PACKAGE sets the default assembly package to either CAMU or PMD. You must set this variable to access the CAMU environment.

  ```
  setenv CV_ASSEMBLY_PACKAGE 'camu'
  ```

- CADDSENABVP enables/disables the automatic creation of a `vp_links` directory for each new part. It should be set to `yes` when you use CAMU.

  ```
  setenv CADDSENABVP 'yes'
  ```

- CV_ASSEMBLY_HANDLE_PART allows you to manipulate the position of a component that is not associated with a part model, using the ORIENT COMPONENT, TRANSLATE COMPONENT, and ROTATE COMPONENT commands. For specific information on creating the default part, see `/usr/apl/cadds/data/am/supplements/README_1`.

  The default part is listed, but any user can override it to access any existing part. You can access the default part using the CVPATH or using a full CADDS style path name.

  ```
  setenv CV_ASSEMBLY_HANDLE_PART 'adhandlepart'
  ```

- CV_UI_DEFAULT_ENV sets default environment when entering CADDS.

  ```
  setenv CV_UI_DEFAULT_ENV 'assembly'
  ```

This eliminates the need to enter `cadds5` with the `-assembly` command line argument. Additional environment variable options are as follows:

.

Table 2-1 Additional Environment Variable Options

Environment	Options
explicit	Explicit modeling menu environment with assembly pulldown.
assyexplicit	Explicit assembly menu environment.
parametric	Parametric modeling menu environment with assembly pulldown.
assyparametric	Parametric assembly menu environment.

- CADDS_MAX_PART sets the maximum number of parts and Adrawings that can be suspended during a single CAMU session. It also controls the number of components that can be imported during a single CAMU session. If not set, the default value of 6 is used. The limit on the maximum number of suspended parts is determined by system resources like memory and swap space. The example below sets the maximum number of suspended parts to 50.

  ```
  setenv CADDS_MAX_PART 50
  ```

- CV_UI_ACT_ASSY sets the default environment when you activate an assembly. The default setting is assyparametric.The setting appears in the Environment: field on the LDM, along with the appropriate value. For the Modeling Environment group, other valid settings are parametric, assyparametric, explicit, and assyexplicit.

  ```
  setenv CV_UI_ACT_ASSY 'assyparametric'
  ```

  For the Eng. Prod. Environment group, the valid variable settings are harness and piping. For the Q.A. Environment group (available only on Sun systems), the valid variable settings are design, dmis, inspection, and programming.

  Refer to the following file for other related variables:

  ```
  /usr/apl/cadds/scripts/templates/.caddsrc
  ```

- CADDSASSOC_UID enables associative HLR and dimensions, and ensures that associativity is properly mapped between the parametric and explicit environments. It also enables parametric positioning. To take advantage of these powerful capabilities, set the following variable in your `.caddsrc-local` file:

  ```
  setenv CADDSASSOC_UID "yes"
  ```

  See "Updating Positions After Modeling Change" on page 6-37 to see a practical application of this setting.

See “Using Pre-Revision 8.0 CADDS Models” on page 6-38 to learn what to do if you have not enabled this variable.

- CV_MAX_ACTIVE_REF sets the limit on the number of active reference assemblies in an assembly session. By default, at most three reference assemblies can be active at the same time in an assembly session.

  ```
  setenv CV_MAX_ACTIVE_REF %max number of active ref assem%
  ```

  When the limit of active reference assembly is reached, you can use the command QUIT REFERENCE to deactivate active reference assemblies.

- CV_POSITION_CONSTRAINT_FEEDBACK enables you to specify a color scheme for displaying parametric positioning constraint feedback in the Modify Position Constraints menu. This menu is opened by the Constrain Component command (see "Modify Position Constraints:" on page 6-17). The defaults are:

  ```
  setenv PC_NOT_SET_COL "Blue"
  setenv PC_SET_COL  "Yellow"
  setenv PC_OVER_DEFINED_COL "Red"
  setenv PC_NOT_CONSISTENT_COL  "Orange"
  setenv PC_NOT_CHANGED_COL "lightgray"
  setenv PC_SATISFIED_COL "Green"
  setenv PC_NO_GEOMETRY_COL "Orange"
  setenv PC_INVALID_COL "Magenta"
  ```

- CV_ASSEMBLY_NO_MODEL enables you to add a component without associating any model to it. If it is not set, or set to "no", the name of the model associated will always be the component name.

  ```
  setenv CV_ASSEMBLY_NO_MODEL "yes"
  ```

- AD_LIST_CONSTRAINTS enables you to list the parametric positioning constraints to a report file window. Set it to "yes" in order to get the listing to a report file window.

  ```
  setenv AD_LIST_CONSTRAINTS "yes"
  ```

- DB_CHECKING specifies the path of the ckCAD executable. It starts Database checking during activating, filing, and exiting of parts. Make sure that the ckCAD executable and its license information does exists at the path specified by `DB_CHECKING` environment variable.

  ```
  setenv DB_CHECKING "/usr/local/bin"
  ```

- The variable `anon`, in the `dfstab` file, enables multiuser access to the ODB_DAEMON process. The `dfstab` file contains list of share commands to be executed while booting. The share commands allows multi user sharing of remote mount disks. Set the variable `annon` to "0" in the `dfstab` file, in order to have multi user access to the ODB_DAEMON process. If it is not set to "0", CADDS can abort.

  ```
  anon=0
  ```

- Setting the USE_CAMU_STATUS_WINDOW variable to "yes" enables you to use the CAMU status window. For better performance during locking and unlocking of components, set this variable to "no". The default value is "yes".

  ```
  setenv USE_CAMU_STATUS_WINDOW "no"
  ```

Please note: The CAMU status window is displayed along with the Assembly Structure Window only. So, to display the CAMU status window you must set environmental variable `AD_C_SERVER_ENABLED` to `no` in addition to setting the variable `USE_CAMU_STATUS_WINDOW`.

Configuring Multi-User CAMU

CAMU can be configured so that multiple users can concurrently access the same assembly. Each user can simultaneously modify different areas of the assembly. They can also see the filed modifications of other users in their active version of the assembly, after issuing the REFRESH ASSEMBLY command.

To configure CAMU for a multi-user environment, do the following:

Determine which system in the network to designate as the DB_DAEMON_HOST. The DB_DAEMON_HOST is set as an environment variable in the CAMU user's `.caddsrc-local` file. It designates a system on the network that is the server for CAMU multi-user assembly access. By default, this is set to the local system.

When a common system is designated by more than one user on the network to be the DB_DAEMON_HOST, CAMU allows multiple users to concurrently access and modify assemblies by connecting them to the active main assembly's warehouse process, called ODB_SERVER.

The ODB_SERVER process is started by another process called ODB_DAEMON when an assembly is activated. The ODB_DAEMON process is started either when the system is rebooted, or manually started by issuing the `/usr/apl/cadds/scripts/db_daemon` command at system level.

By setting the DB_DAEMON_HOST in the `.caddsrc-local` file, you are determining on which system the ODB_DAEMON process creates the ODB_SERVER process for the assembly you are activating. This happens whether or not you intend to run multi-user CAMU.

For example, in `.caddsrc-local`:

```
setenv DB_DAEMON_HOST 'systemname'
```

Choosing a Host System for DB_DAEMON_HOST

Although you can choose any system in the network as the host system for DB_DAEMON_HOST, be sure to evaluate the considerations given on the following pages to achieve optimum performance and efficiency of the system:

Performance Considerations

Although you can designate a CADDS user's system as the DB_DAEMON_HOST, the user might experience some reduction in system performance.

The following factors determine the degree of performance reduction, if any, relative to CAMU usage:

- CPU configuration
- Memory (RAM) available
- Amount of concurrent users accessing active assemblies
- Size of the active assemblies
- Amount of separate assemblies
- Degree of activation/filing activity in active assemblies

Network Considerations

Performance of multi-user CAMU can be affected by the network configuration. If the ODB_SERVER process is running on a remote system, runtime performance could be degraded due to the network problems.

If the DB_DAEMON_HOST is set up on a system separate from the system that has the assembly directory or part library directories, performance can be affected when activating or filing assemblies. This is because the network would have to be traversed when either reading in data from the parent databases to the warehouse process, or writing to the parent databases during filing operations.

To achieve optimum performance of a CAMU multi-user configuration, set the DB_DAEMON_HOST on a fileserver type system that also contains the common assembly directory and assembly library parts directory.

NFS (Network File System) Prerequisites

For remote NFS mounts, each user's local system mount point to a common assembly directory on a remote system must be the same on all systems. The absolute path name of an assembly must be the same for all users who concurrently access that same assembly.

Example:
The common assembly directory exists on system "server" as `/usr2/parts/assys`.

The local systems have to mount `server:/usr2/parts/assys` from the same mount point directory, for example,
`mount server:/usr2/parts/assys`

Or, the local systems can use a mount point that is not necessarily the same directory name as the server's file system:

```
mount server:/usr2/parts/assys /users/assyparts
```

As long as the mount points on all the local systems accessing the common assembly directory are identical, users configured for multi-user CAMU can concurrently access the assembly databases.

If the server's assembly directory is accessed from different mount points on the network, although the assembly database is accessible through different mount points through NFS, the absolute path name is different. The result would be that each user would essentially be activating different assemblies, even though they have NFS access to the same assembly directory.

Example:
User 1 has the server's assembly directory mounted as follows:

```
mount server:/usr2/parts/assys /users/assys
```

User 2 has the same server's directory mounted on another system through a different mount point:

```
mount server:/usr2/parts/assys /usr2/parts/assys
```

When each user changes to the mounted assembly directory, their current working directory is different:

```
User 1:cd /users/assys
pwd = /users/assys
User 2: cd /usr2/parts/assys
pwd = /usr2/parts/assys
```

As a result, each user can list the same assemblies from their respective mount point directory. However, when they attempt to activate the same assembly from CADDS, user 1 activates:

```
=users.assys.assembly
```

and user 2 activates:

```
=usr2.parts.assys.assembly
```

Both users get a message saying "Activating Old Assembly", but only one user is active in the correct assembly.

CVPATH Prerequisites

Each user's CVPATH entry in the `.caddsrc-local` file referencing the common assembly directory must be exactly the same for all users.

As is the case with NFS mounts, the manner in which the common assembly is accessed in each user's CVPATH must be the same for multi-user CAMU. The most straightforward way to ensure this is to reference the mount point of the assembly directory in the CVPATH, for example:

```
setenv CVPATH $HOME/parts=C:/usr2/parts/assys:
```

Or, if you want the assembly directory to also be the Create directory:

```
setenv CVPATH /usr2/parts/assys=C:$HOME/parts:
```

Please note: Do not use the $HOME variable for the common assembly directory. Each user may have a different home directory.

The restrictions that apply to the CVPATH for assembly databases do not apply when accessing the component models of the assemblies. As long as the path specified in CVPATH references the directory level in which the component models exist, the assembly finds them, regardless of how they are referenced through the NFS mount.

If you have problems accessing the components of an assembly, choose Verify Object and find the name of the component part that the assembly database is referencing, then provide a directory in the CVPATH that points to this directory.

Example:
User 1 and User 2 Active Assembly=`usr2.parts.assys.assembly`
Assembly parts actually reside on the server in the `/users/libparts` directory.
User 1 has a parts directory mounted as `server:/users/parts`.
User 2 has it mounted as `/usr2/libparts`.
If both user 1 and user2 have paths to `/users/parts`, and `/usr2/libpart` respectively, then the assembly finds the parts.

Software Prerequisites

The DB_DAEMON_HOST system and the local systems must have the same revision of CAMU (specifically, the ODB_DAEMON and ODB_SERVER executable files under `/usr/apl/cadds/bin`) to run multi-user CAMU across the network.

Other System Considerations

If you do not have write permissions to an assembly, the system does not allow you to activate it.

To set up a configuration that allows write access to a common assembly directory and/or library parts directories, give users the same group id in the `/etc/passwd` file. Next, insert the `umask 2` command in the `.login` or `.cshrc` file for each user. This turns on read/write permissions on files created by the user and allows any member of the group to activate and modify files created by any member of the same group.

Sample Procedure to Configure Multi-User CAMU

Follow this procedure to configure multi-user CAMU.

1. Establish which system on the network is to be the DB_DAEMON_HOST.
   - Is the system accessible to all CAMU users on the network?
   - Is the same revision of CADDS installed on the system?
   - Is there an ODB_DAEMON process running on the system?
   - If the DB_DAEMON_HOST system is not the same system that the assemblies “live” on, can it access the system(s) where the common assembly directory and parts exist?

The system designated as the DB_DAEMON_HOST for this example is a file server named `server`. It contains the assembly directory, `/usr2/parts/assemblies` and the assembly part library directory, `/usr3/cadds/libparts`. The local systems accessing the DB_DAEMON_HOST are called `system1`, `system2`, etc.

2. Establish the mount points from the local systems to the file server and test them on the local systems.
   - Are the assemblies and parts directories mountable on the file server?
   - When the mount point directories are created on the local systems and mounted on the file server, you can list the files contained in the mounted directories when you `cd` to them.

Example:
On local systems:

```
mkdir -p /usr2/parts/assemblies /usr3/cadds/libparts
mount server:/usr2/parts/assemblies /usr2/parts/assemblies
mount server:/usr3/cadds/libparts /usr3/cadds/libparts
cd /usr2/parts/assemblies
```

`pwd` (should output `/usr2/parts/assemblies` on all local systems)
`ls` (are the files in the directory listed out?)
Follow the same procedure to check `/usr3/cadds/libparts`.

3. Configure user `.caddsrc-local` files for multi-user CAMU.
   - Set the DB_DAEMON_HOST variable to the system server. For example:

     ```
     setenv DB_DAEMON_HOST 'server'
     ```

   - Set the CVPATH variable to access desired assembly and parts directories.

Example 1:

In this example the `.caddsrc-local` files of the users are set up so that the assembly directory is also the Create directory. The directory containing library parts is also accessible in the user's CVPATH. Any parts in the user's local parts directory (`$HOME/parts`) are accessible as well.

```
CVPATH /usr2/parts/assemblies=C:/usr3/cadds/libparts:$HOME/parts:
```

Example 2:

In the following example, the CVPATH is set up so that each user's local parts directory is the Create directory, but the common assembly directory and the library parts directory are also accessed. The order of the path is important. If duplicate part names exist in different directories, the part in the directory that appears first in the CVPATH is accessed.

```
CVPATH $HOME/parts=C:/usr2/parts/assemblies: /usr3/cadds/libparts:
```

Example 3:

If the parts that exist in the `/usr3/cadds/libparts` directory are to be accessed before any local parts that exist with the same name, and you still want your local directory to be the create directory:

```
CVPATH /usr3/cadds/libparts:/usr2/parts/assemblies:$HOME/parts=C:
```

Please note: The CVPATHs can vary from user to user as long as the common assembly directory is accessed with the same absolute path.

Testing the Multi-User Configuration

Once you have completed the CAMU multi-user configuration, test the configuration to ensure that multi-user access to assemblies works.

1. Verify that all NFS mounts are established correctly.
   a. Change to the mounted directories.
   b. Verify that all users' absolute path names to the mounted directory are identical by typing `pwd`.
   c. List any existing directories and check that the ownership/permissions allow access to group members.
2. Log in to the user accounts and enter CAMU on local systems. The assemblies (if existing) should be displayed.

   If there are existing assemblies in the mounted file system and they do not display in CAMU, do the following:
   a. Check the absolute path name for the active directory displayed in the LDM for accuracy.
   b. If the assembly directory is not the create directory in CVPATH, either select the Active Directory, or Display all Files in CVPATH from the LDM menu.
   c. Check to see if the remote file system is mounted correctly.
   d. Verify permissions on the assembly directories.
   e. Activate either an existing assembly or a new one.
   f. If there are problems activating a new assembly or an existing assembly, check the 'Troubleshooting' section that follows.
3. Add a component to the root node of the assembly.
4. File the assembly.

   This test establishes that the assembly database can be written by at least the user who created it.
5. Activate the test assembly from the other user accounts.

   Log in to the other network systems using the accounts that run multi-user CAMU, and follow steps 1 and 2 above. Activate the assembly which is still active from the first user, but specify a new Adrawing name in the Adrawing Name field of the Activate Assembly menu.

   Each user should be able to activate the tree and see the root node and the additional component which was added by the first user.

6. Add a component from the second user, file the assembly, and issue REFRESH ASSEMBLY from the first user's session.

   The first user should be able to see the component added by the second user.

7. Follow the preceding steps for all additional multi-user accounts on the remaining networked systems.

   For each subsequent user account, activate the currently active assembly, add a component, and issue REFRESH ASSEMBLY for all previous users. Assembly structure should show the updated versions of the tree with each user's components.

8. Verify that graphics display for component model geometry.

- Issue ACTIVATE MODEL for the component that each user added.
- Insert model geometry into the active model.
- Use the Activate Old Adrawing property sheet to activate the suspended Adrawing for each user.
- Select the root node from the tree and use the View component property sheet (with Detail View enabled). Each user should see only the geometry he has just added to his model(s).
- Issue FILE ASSEMBLY for each user.

  Because FILE ASSEMBLY automatically performs a refresh (REFRESH ASSEMBLY), each user should now be able to see the geometry for all the components in the assembly. However, due to the sequence of filing, a REFRESH ASSEMBLY may be required after all users file their active main assembly to display the current version of the assembly.

Troubleshooting

Follow the tips given below when you encounter errors configuring multi-user CAMU.

Activating a New or an Existing Assembly

These are the common errors you come across while activating a new or an existing assembly.

- Warehouse Errors or RPC Errors or Error Code 22

  No ODB_DAEMON process running on the DB_DAEMON_HOST system.

  Check the system designated using the following command:
  `ps -ef | grep ODB_DAEMON.`

  If there is no process, manually start it by issuing
  `/usr/apl/cadds/scripts/db_daemon` on the host system.

- Assembly is already active by another user on a DB_DAEMON_HOST other than what you have specified.

  To resolve this, switch your DB_DAEMON_HOST to the system specified by the other user.

- LOCK or TEMP files exist on the assembly database file (_db file) from an abnormally exited session.

  To correct this, inspect the assembly database file and the ownership of any LOCK or TEMP files. Ensure that no other user is currently active in the assembly and then remove the LOCK and TEMP files.

- Revision conflict of ODB_DAEMON or ODB_SERVER code.

  Verify that the same revision of CADDS, specifically /usr/apl/cadds/bin/{ODB_DAEMON, ODB_SERVER} is installed on the remote DB_DAEMON_HOST.

- You cannot lock components of an assembly that are locked by other users. You can determine who has locked any given component by selecting it and using the Verify Object property sheet.

- If you activate an assembly that was exited abnormally, and then attempt to lock components that you had locked in the previous session, the system displays a message indicating that the component is locked by another user. Correct this at the OS level by removing the LOCK and TEMP files in the model's subdirectory.

Single User Environment

In a single user assembly environment, the common cause of a problem is that there is a default ODB_SERVER process connected to the assembly from the previous session.

To resolve this, do the following:

1. Find the associated ODB_SERVER process using the ps command.

   The associated assembly name is appended to the ODB_SERVER process in the output.

2. Kill the ODB_SERVER process with the UNIX kill <pid> command, and reactivate the assembly.

Please note: Do not use the kill -9 command to kill the ODB_SERVER process. This may create problems with the ODB_DAEMON, which will prevent you from activating the assembly previously activated by this process. Only the superuser can kill the ODB_SERVER process.

3. Delete all `*LOCK` and `*TEMP` files within the assembly's directory.
4. Activate the assembly.

Please note: Only the superuser can kill the ODB_SERVER process.

Terminating the ASSEMBLY_DESIGN and ODB_SERVER Process

Do not use the `kill -9` command to terminate the ASSEMBLY_DESIGN or ODB_SERVER process. This is especially important if multiple users are working on the same assembly. If one user issues `kill -9` on the ASSEMBLY_DESIGN process, it does not detach from the ODB_SERVER process and prevents other users from continuing their work on the assembly. If you must terminate this process, use `kill` without the `-9` option.

Multi-User Environment

In a multi-user environment a problem can occur when the active main assembly's ODB_SERVER process has not properly terminated a particular user's session, but the assembly is still active by other users.

To resolve this condition, all users currently active in the assembly must exit the assembly to terminate the ODB_SERVER process. The ODB_SERVER process should then end normally and all users should be able to activate the assembly and resume work.

Other issues that can occur in the multi-user environment are as follows:

- The CADDS command window displays "`Cannot Activate Assembly...`" messages. These message could be displayed due to any of the following reasons:
  - User does not have correct permission to the `create` directory or existing assembly
  - Problems with the `/usr/tmp` directory, such as insufficient permission, lack of disk space
  - Incorrect path to mounted assembly's file system
- System displays a "`Activating Old Assembly`" message when activating the assembly that is currently active by another user. However, when the assembly is activated, it only displays the root node without the component added by the previous user.

  This behavior is caused by users having different `CVPATH`s which point to the same assembly directory, but access it through different absolute paths.

To resolve this condition, modify the user's `.caddsrc-local` files so that the `CVPATH`s pointing to the common assembly directory are identical.

- System displays a `"Cannot Activate Assembly. Assembly is in use by another user"` message.
  - Check the DB_DAEMON_HOST environment variable in the user's `.caddsrc-local` to make sure it points to the common system designated as the DB_DAEMON_HOST by the active user.
  - Check the mount point systems concurrently accessing the assembly. They must be identical.
- The following errors are displayed if a user activates an assembly which has been activated as a new assembly by another user, but has not been filed yet.

```
ERROR: This assembly is being created by another user.
It cannot be activated until it has been filed.
ERROR: Errors have occurred in the warehouse process.
ERROR: There is a problem activating the assembly tree <assembly
name>
You have to reissue ACTIVATE ASSEMBLY.
ERROR: Errors have occurred in the warehouse process.
 Assembly Design Version 2.0
 Activating Assembly Design Environment
Command only valid while there is an active part.
Command only valid while there is an active part.
```

To resolve these errors, the user who originally activated the assembly must file the assembly.

- The following message is displayed when you activate an assembly using the DEFAULT Adrawing (or any Adrawing) that has already been activated by another user. The Adrawing is activated for read access only.

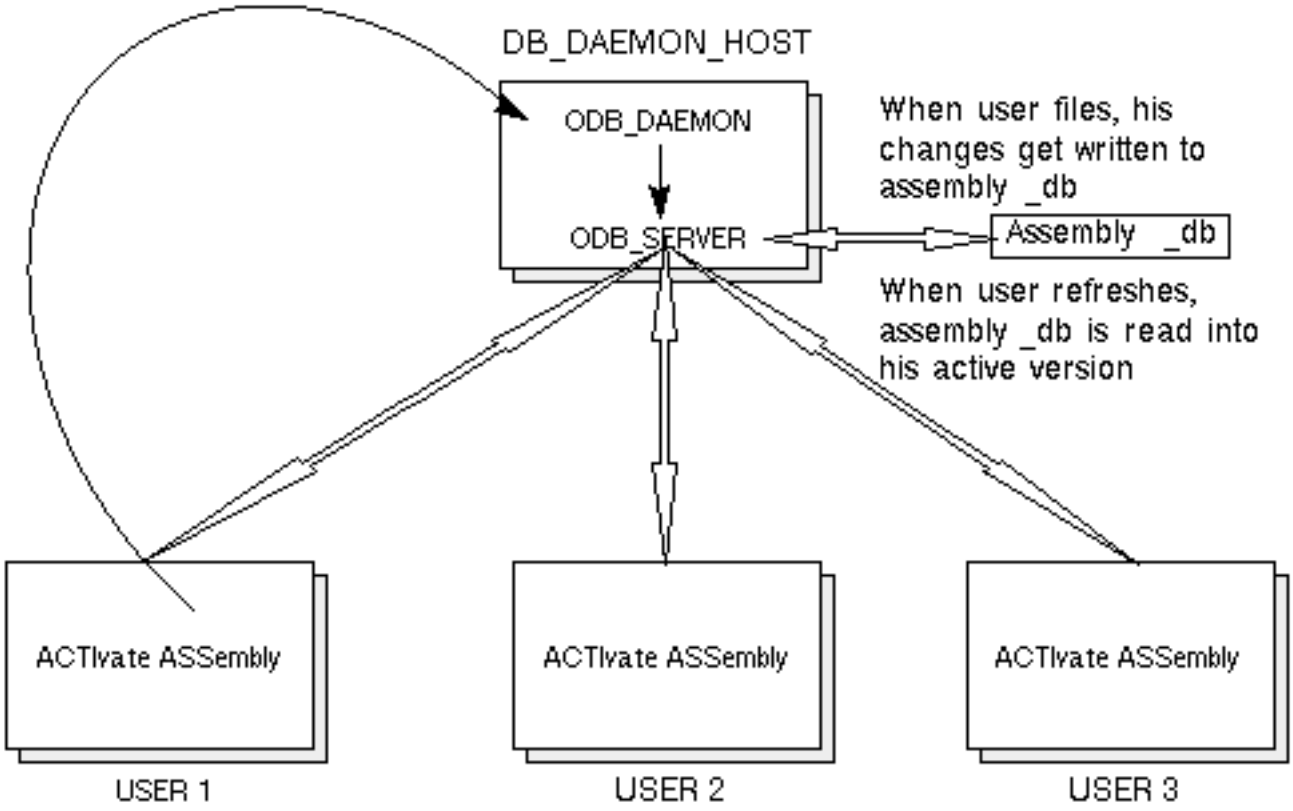
```
Activating OLD part.
This is a CADDS5 DESIGN.
The file is in use.
Part is activated but may not be filed under this name.
The Assembly Drawing is read-only.
```

Please note: If graphics are not visible for all of the components, check the following conditions:

- If in the Parametric environment, are all layers echoed?
- Is there a view defined for the active Adrawing?
- Does the model have a current explicit graphics file (`_gr`)? If not, either run `update_gr` or activate the model in the Explicit environment, run the Render View or Change Renderview command with any of the shade or HLR rendering options, and file the part.

Refer to the multi-user diagram shown below.

Figure 2-1 The Multi-User Process



ODB (Object Data Base) Administration Tool

The Unix level command `odb_admin` (`apl/cadds/scripts`) deletes users from the `ODB_SERVER` list who have crashed out of their assembly. In the event of an abnormal termination of a multi-user CAMU session, the tool can detach specified users and release any associated locked components. This avoids having to restart the `ODB_SERVER` for everyone. The command simply prompts “Enter the full path assembly name” and then clears the non active users.

The `odb_admin` tool also allows users to connect to the assembly’s ODB_SERVER process.

To run the tool, execute the `/usr/apl/cadds/scripts/odb_admin` script on any supported UNIX workstation. The required server host is determined through the `.caddsrc` or `.caddsrc-local` file.

The user is prompted for the full path of the assembly name. The assembly name is entered either in UNIX style or CADDS style, without the `_db` extension. For example, either `/users/jdoe/parts/engine` or `=USERS.JDOE.PARTS.ENGINE` are acceptable.

Upon successful connection to the ODB_SERVER, the tool lists all users attached to the assembly in the following format:

USER ID	USER NAME	SYSTEM NAME	PROCESS_ID	USER TYPE
1	jdoe	reggae	8539	CAMU User
2	bgates	windfall	3456	API User
3	abc	system1	1550	CAMU User
4	def	system2	2545	CAMU User
5	ijk	system3	3232	CAMU User
6	pqr	system4	4342	CAMU User
7	xyz	system5	5478	CAMU User

The tool prompts for the user ID of the non active session. After checking that the selected session is not in use, it releases the appropriate locks and detaches the user process from the server. Following this action, the other users have access to the components that were locked.

Arranging Projects Using CVPATH

In a large production environment, it may be desirable to separate various aspects of assembly design into separate projects or zones.

You can organize various directories or remote file systems that contain parts and assemblies for individual projects by setting the CVPATH variable in the `.caddsrc-local` file. After organizing projects with CVPATH, after you enter CADDS, you can then select the project and its accompanying assemblies and parts, by clicking on the CVPATH menu bar on the Assembly LDM.

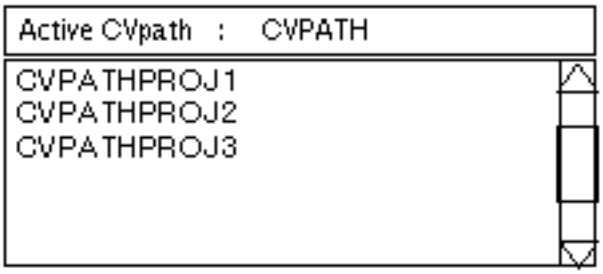
Example of setting multiple CVPATHs:

Your company makes automobiles. Within the production environment, you have projects setup for engine design, frame design, and body design. However, the users on each team need to work on all aspects of the automobile design.

You can set multiple CVPATH environments in your `.caddsrc-local` file to automatically access the correct set of file systems containing the assemblies and parts for each project.

```
setenv CVPATHPROJ1 /users/parts/engine:$HOME/parts=C:
setenv CVPATHPROJ2 /users/parts/frame:$HOME/parts=C:
setenv CVPATHPROJ3 /users/parts/body:$HOME/parts=C:
```

With these CVPATHs set, after you enter the LDM environment, click on the CVPATH menu bar. This displays a runtime list of all the available CVPATHs.



If you are working on several different assembly projects, and you do not need access to every part for a particular project, organizing projects with CVPATH reduces the time taken to generate the LDM listing.

Data Management

You can use a data management tool, such as Vault, to perform checkin, checkout, and revision control on the assembly and all its constituent components. You can use the Vault and CAMU together from within EPD.Connect.

Assembly Database Items

The assembly directory, which equates to the name entered on the Activate New Assembly or Activate Old Assembly property sheet, must be tracked by the database management system.

The assembly consists of the following components:

- The assembly file itself (`_db`)
- All Adrawings (CADDS parts subordinate to the `_db` and used to view, position, and document the assembly geometry) in the assembly directory, such as `default`
- The `root` model (if there is one) in the assembly directory
- Links to models associated with components in the assembly structure (`_db`)

Model Items

Each model associated with components within the assembly structure must be addressed by the database management system. The model directory, which equates to the name entered in the Activate Model property sheet (or outside CAMU, on the Activate Part property sheets) contains some or all of the following items that must be managed:

- The part database itself (`_pd` and `_fd`).
- The explicit graphics file (`_gr`) containing the drawing and model graphics data along with the tessellation data.
- The `vp_links` directory.

  This directory must exist to house the files generated by the VIEW COMPONENT command. These files are removed when you unview or exit the assembly. In case of abnormal shutdown, you should manually delete these files.

Locking Models

Do not use the Model option when locking (LOCK COMPONENT) unless you want to prevent other users from activating the associated model. When you use the Model option, CAMU creates a `LOCK` and `TEMP` file in the associated models' directory. In the event of an abnormal shutdown, you should delete these `LOCK` and `TEMP` files. This also means that it is worth quitting any suspended models after you file them.

Archiving the Assembly

Ensure that you archive the following items for any assembly:

- The assembly database (`assembly_name/_db`)
- All Adrawings (`assembly_name/adraw_name/_pd` and `assembly_name/adraw_name/_fd`)
- The DDRAW of the Adrawing named `default` (`assembly_name/DEFAULT/_gr`)
- All models associated with component classes in the assembly or product structure (`model_name/_pd` and `model_name/_fd`)
- The explicit graphics file (`_gr`) for all models associated with component classes in the assembly or product structure (`model_name/_gr`), which contains the drawing and model graphics data along with the tessellation data.
- The `vp_links` subdirectory of each model (`model_name/vp_links`) see "vp_links Directory" on page 2-24
- All explicit figures (nodal figures) used by or within models or Adrawings (`figure_name/_nfig` and `figure_name/_gr`)
- All procedure files (parametric) and execute files (explicit)
- All applicable features databases
- All NC toolpaths (`model_name/*.jcf`)
- All model-specific CVMAC routines

CADDS Drawing Files

All the CADDS drawing and model graphics information is stored in the explicit graphics file (`_gr`). The `_gr` file is subordinate to the CADDS part (model) that was active at the time you issued the ACTIVATE DRAWING command. You must consider archiving the `_gr` file of each model in your assembly when you archive the assembly and its models so that you can view the component in CAMU.

Please note: Whenever you activate and save a part, the size of the explicit graphics file (_gr) file may increase or decrease by up to 1%. The information in the file is restructured or reordered every time you save the part. This is an expected behavior when compressed data is read from and written to a file.

If the _gr file of your model is missing, the VIEW COMPONENT command dynamically generates the graphics data for that model by reading the _pd file of that model. However, this can adversely impact the performance of the VIEW COMPONENT command. Additionally, to view the model in any of the shade or HLR rendering modes, the _gr file must contain the tessellated data for the solid or surface geometry in that part. To create the tessellated data in the _gr file, activate the part in the Explicit environment, run the Render View or Change Renderview command with any of the shade or HLR rendering options, and file the part without specifying the NOSHADE or NOTVF option.

vp_links Directory

A vp_links subdirectory is created when you activate a part using either ACTIVATE PART or ACTIVATE MODEL, provided you have the following line enabled in your .caddsrc or .caddsrc-local file:

```
setenv CADDSENABVP 'yes'
```

The vp_links directory houses files that are created by the VIEW COMPONENT command. These files exist only for the duration of the time that the associated model geometry is viewed. They are deleted when you unview or exit the assembly.

This directory is required for model viewing in CAMU. For this reason, you should archive the vp_links directory of each model in your assembly when you archive the assembly and its models.

Aborting a Command

<Control e> aborts a command. It will get you out of most context switching problems where the error message Command not valid at this time has appeared.

Crash Recovery

If you or another user who is also active in the assembly and there is an abnormal shutdown, follow this crash recovery sequence:

1. Run a `slay` script to kill your CADDS processes.

   Please note: You are provided with `/usr/apl/cadds/scripts/slay`.

2. If there are others still working concurrently in the assembly, see "ODB (Object Data Base) Administration Tool" on page 2-19 before proceeding.

3. Remove LOCK and TEMP files for:

   - All suspended models
   - All suspended Adrawings
   - The active part (Adrawing or model)
   - The active assembly database
   - All models that you locked (LOCK COMPONENT) using the MODEL option

4. Remove all files in each applicable `vp_links` directory.

   Please note: Junk files will be in the `vp_links` subdirectory of each model that was viewed (VIEW COMPONENT) at the time of the abnormal shutdown. Delete the files, not the `vp_links` directory itself.

5. Kill the ODB_SERVER process associated with the assembly using a straight `kill` command.

   - Make sure you:

     Do not use `kill -9`. This can corrupt the assembly.

     Do not kill the ODB_SERVER process if anyone is still in the assembly.

     Only kill the ODB_SERVER process that belongs to the hung assembly. Use a `ps` command such as `ps -ef` to see both the process ID and the assembly name associated with the process.

6. Kill the CALLBACK_SERVER processes. There will be a CALLBACK_SERVER process for each reference assembly that was active when the abnormal shutdown occurred.

   Please note: The `slay` script may kill CALLBACK_SERVER processes.

Chapter 3

Creating a New Assembly

This chapter describes how you can create a new assembly, add components and subassemblies to the assembly, and optionally view and position associated geometry. The topics are:

- Creating (Activating) a New Assembly
- Building the Assembly Structure
- Adding a New Component to the Structure
- Adding a Family Member of a Master Model
- Multiple Occurrences of the Same Item
- Relations Between Components in the Structure
- Adding Existing Assemblies to Your Assembly
- Creating a New Assembly by Exporting

Creating (Activating) a New Assembly

These instructions show you how to create (activate) a new assembly. This activated assembly uses existing parts that contain geometry and new parts for which you create geometry later. For existing parts, view and position the parts on your screen. Your assembly can also incorporate an assembly that exists outside your active main assembly (called a reference assembly) as a subassembly.

Using the Default Adrawing

An Adrawing is both a drawing and a special CADDS part. It allows you to:

- View the model geometry associated with the components of the assembly
- Define the view state (scale, orientation, location, and size) of component instances in the assembly
- Use the viewed geometry to establish the position and orientation of component instances
- Determine which components are viewed in a particular Adrawing

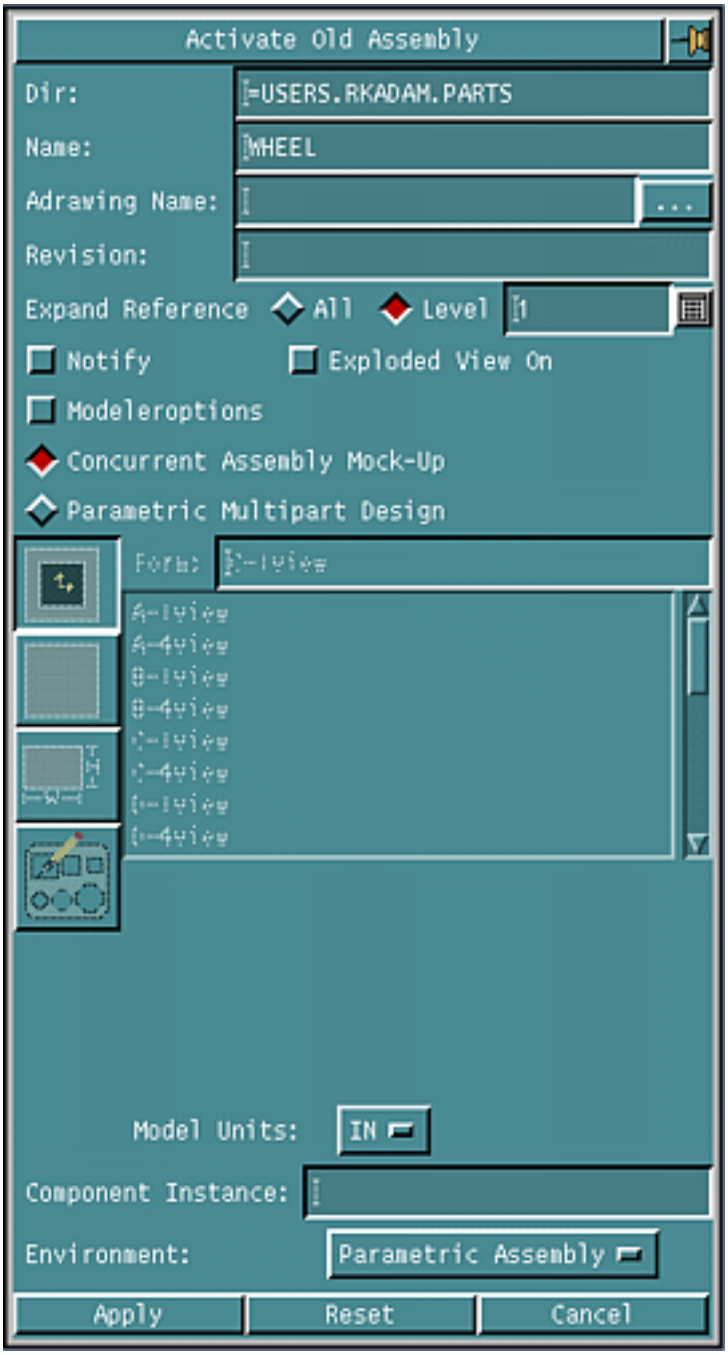
While creating a new assembly, you can specify an Adrawing other than the DEFAULT Adrawing. The assembly with the DEFAULT Adrawing is first activated, and then the specified Adrawing for the active main assembly.

Steps to Activate a New Assembly

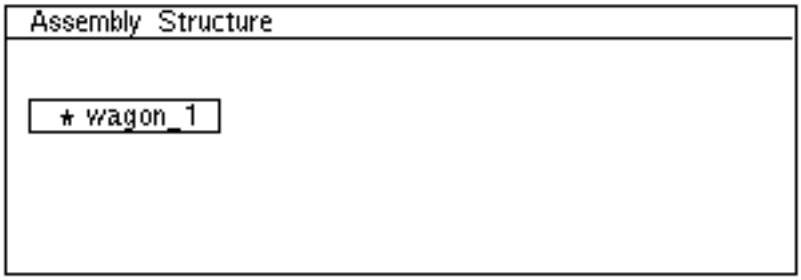
To activate an assembly, do the following:

1. From the LDM menu, select Activate New Assembly and Adrawing. The Activate New Assembly and Adrawing property sheet appears.
2. Choose the CAMU toggle switch if it is not already selected.
3. Enter the name of the new assembly in the Assembly Name field, for example, `wagon`.
4. Enter the name of the Adrawing you wish to activate. If you do not specify a name the system activates the default Adrawing.
5. Use the default Environment or select the environment from the menu.
6. Click Apply to process the property sheet.

Figure 3-1 Property Sheet to Activate a New Assembly



The name of the assembly appears in the Assembly Structure window. This is the root component of the assembly. An underscore (_) sign and instance number is appended to the assembly name. The asterisk (*) indicates the component is locked by you.



Building the Assembly Structure

You can build the assembly structure by adding components or external assemblies to the active main assembly. Components may or may not be associated with piece parts (models). You can position the component or assembly instance in the active main assembly and view it if it has an associated model.

While creating a component, you must specify:

- A name for the new component (class name only, not instance number)
- The component to which you want the new component attached (parent)

If you do not specify a parent component, the new component(s) are attached to the root component.

Please note: If you have set Select Assemble On, see "Automatic Assembly" on page 6-32, all parametric positioning constraints get solved when you add a component or an assembly.

Typical Structuring Approaches

Typically you design your CAMU assembly structure by functional groupings. The three most prevalent scenarios are:

- Create a master assembly containing the complete product assembly that all users can access (manipulate, view, query).
- Create a separate assembly for each major functional grouping (main subassembly branches). For example, a vehicle would contain a body, chassis, and electrical assembly.
    - The project manager can assemble the entire product by adding each of these functional assemblies to the master assembly database as reference assemblies.
    - The users within each assembly can see the entire assembly (or specific portions) by adding a reference instance (adding a reference assembly) of each of these top-level assemblies into their own assembly.
    - Prevent nesting and looping reference assemblies.
- Create trees by geographic area, and in these zones break down the trees by functional groups. One technique is to:
    - Divide the product (car, ship, airplane, etc.) into a series of zones based on factors such as area complexity, people involved, geographic locations for design and manufacture, an so on.

- Create zone trees to group all the models in the geographic zone together.
- Use a module type approach to construct trees from the functional areas but divided by geography.

For example, an engine manufacturer could have a number of trees for the brakes module, depending on how many zones the brakes module was contained in.

Guidelines for Creating the Assembly Structure

- Build one master assembly in which all users shall work concurrently. Each functional group requires its own subassembly branch in the master assembly.
- Structure the assembly based on functional groupings. For example, a chassis subassembly would include the brake system subassemblies.
- Assign an owner to each major subassembly branch in the master assembly. The owner shall generate, input, and maintain the subassembly branch. If one branch contains components from several designers, a single owner is still required.
- Include all high priority parts in the assembly structure. High priority is determined by the group or project team as either package-critical, weight-sensitive, and so forth.
- Have at least read access to all associated models in your assemblies.
- Associate CADDS models to leaf nodes only; not to subassembly nodes.
- Use a standardized naming convention for both component names in the assembly structure as well as your CADDS models. Should you change a component class or a model name, communicate that information to the group. The master assembly root node, as well as the top level node of each subassembly, must also conform to the naming conventions.
- Ensure that all models associated with components in a branch belong to the same function group.
- For each component class that does not have a model associated with it (such as subassembly components), use the No Model option on the Change Component property sheet. Otherwise a search through the CVPATH for the associated model to that component will be performed, even though one may not exist.

Special guidelines are listed below:

- If the subassembly branch contains models from two or more component areas, ownership of the branch must be decided and agreed upon.
- For model geometry of a current design and/or another possible design, the new proposal must also be shown in the assembly structure.

- Where subassembly branches contain derivative models, important derivatives based on worst-case conditions must be shown in the assembly structure.

Naming Conventions

You should establish and adhere to a standardized naming convention. Each of the following requires such a convention:

- Models
  - The model name should match the actual part number it represents.
  - Derivative parts created using Insert Ppart should be named to indicate the source model's name.
- Drawings

  Drawing names should adhere to either your existing drawing name conventions (for example, `sheet1of3`) or some other agreed upon convention.
- Assembly database

  A sample naming convention could consist of three or four distinct items - an assembly type abbreviation, a project name or code, a four-digit functional group code, and, optionally, a brief descriptor. Your convention could be simpler or more complex.
- Adrawings

  You can (and usually do) have multiple Adrawings in an assembly. Use a descriptive method for naming Adrawings, for example `Exploded-All`, `Design`, `Section AA`, `Sheet1of2`, or `Sheet2of2`.

  You can activate additional drawings in an Adrawing. Again, establish a naming convention and adhere to it. For example - `1of2`, `2of2`.
- Assembly structure components

  A general guideline is to use descriptors as component class names and actual part numbers for the associated model names.

Ways You Can Add Components or Assemblies

There are three ways you can add components or assemblies to the active main assembly:

- Add a component (or assembly) that may or may not have an associated part or model. The component appears in the assembly structure. If the component has an associated model, and you do not position the component, it defaults to Layer [0] and TOP Cplane.

- Add a component (or assembly) that has no associated part/model, and specify the position(s) where the component (and its multiple instances) appears in the assembly. Later, when you activate the model to build its geometry, it appears appropriately placed in the assembly.
- Add a component (or assembly) which already has an associated model; view the model geometry and position it in the assembly.

Please note: You can add a standard model or a member of a master model "Options on the Property Sheet" on page 3-11.

Please note: When orienting components that have no associated CADDS model, the following message appears in the Text window. For example, you may want to orient a subassembly whose parent component has no associated model.

```
WARNING: The model part 'partname' does not exist
         You are now using the handlepart.
```

Adding a New Component to the Structure

You can add one or more instances of a component using the icon from the Component/Assembly task set or by choosing the Add Component option from the Assembly Structure window pulldown menu. You can view and position the component in the graphics window.

A sample Add Component task is illustrated below. See page 3-11 to learn more about each property sheet option.

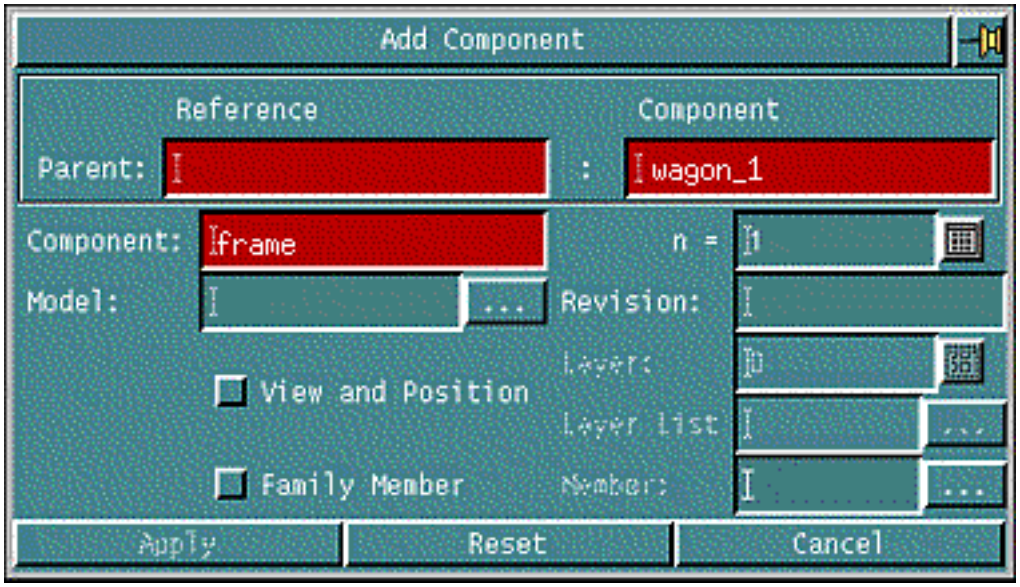
1. Choose the Add Component option from the Component/Assembly task set. The Add Component property sheet appears.
2. Select the desired parent component from the assembly structure to automatically enter its name in the Parent field on the Add Component property sheet.

OR

3. Select the parent component to which you will add the new component. Select the parent component from in the assembly structure.
4. Press and hold the right mouse button and choose Add Component from the resultant pulldown menu. The Add Component property sheet appears.

For example, if you select a component named `wagon_1`, press and hold the right mouse button, then choose Add Component from the pull down menu, the following property sheet appears with the parent name `wagon_1`.

Figure 3-2 Add Component property sheet



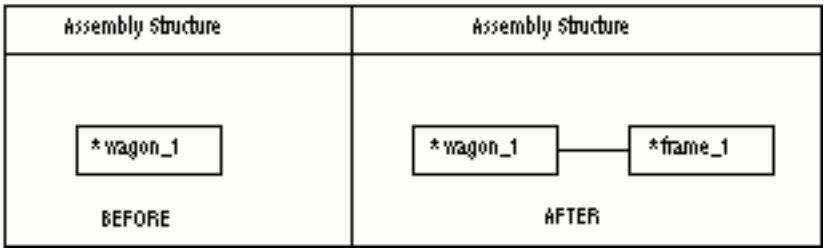
Please note: If a parent is not specified, the component you add is attached to the root component.

5. Select in the Component field.
6. Type a component name, for example, `frame`. When you apply the property sheet a component instance named `frame_1 is added` to the `wagon_1` component.
7. To add more than one instance of the component, place the cursor in the n = field and click the left mouse button. The calculator appears for you to specify the number of component instances you want to create.
8. If the model associated with the component has a different name than the component name, type the model name in the Model field. See page 3-11 to learn more about using the Model option when associating a component with a standard or master model.

Please note: If no model name is specified, the component is automatically associated with a model of the same name. You can, if desired, add a component without any model associated with it, by setting the environmental variable CV_ASSEMBLY_NO_MODEL to "no". ".caddsrc-local File Information" on page 2-3 for more details on setting this variable.

9. If desired, specify a revision number in the Revision field. This field allows you to access different versions of the model part associated with the component. For example, if the revision number is `2` and the model part name is `frame`, the actual model part name is `frame-2`.
10. To update the assembly, click Apply. The assembly structure gets updated.

Figure 3-3 Changes in the Assembly Structure on adding a component



Please note: If the new component represents a model that has not yet been built, you can enter the name of the model that is to be built later. You can add a component without associating it with a model.

In case you specify the View and Position option while adding a new component which is not associated with an existing model, the system accesses the default CADDS part called `adhandlepart` that resides in the CADDS create directory. This is a CADDS part that you create and point to using an environment variable in the `.caddsrc` or `.caddsrc-local` file. It provides you with model space until the actual part is built. See ".caddsrc-local File Information" on page 2-3 for more information about setting up the adhandlepart.

Options on the Property Sheet

The Add Component property sheet options are described below:

- Parent - Specifies the name of the component instance to which the new component will be added. The new component will become a descendent of the parent specified. Type the parent name or select it from in the assembly structure, using the middle mouse button.
- Component - Specifies the class name of the component you wish to add. Do not type the underscore and instance number, they will be created automatically when you add the component.

Please note: The component class name and the model name can optionally be identical. However, the Component class name is often used as a descriptive term while the model name represents a part number or other company-specific method of naming actual CADDS parts.

- n - Specifies the number of instances of the component class that you wish to add.
- Model - Specifies the name of the CADDS part to associate with the component class name. The part (model) can exist currently but does not need to. The Model option is only valid when adding a new component class.

Please note: If you wish to use the Family Member option, the model name specified must be a Master model as defined using CADDS Family of Parts functionality.

- Revision- Specifies the revision of the associated model. This specifically pertains to a dash letter or dash number following the model name on disk.
- Layer - Allows you to specify a value by which to increment layer display. For example, if the model geometry resides on layers 3 and 30 and you specify a Layer value of 6, the model geometry appears on layers 9 and 36 in the active Adrawing. The default value is 0.
- Layer List - Allows you to specify a range of viewing layers for the associated model geometry. For example, the associated model may contain geometry on layers 2 through 5 and layers 10-15 that you wish to see, while you do not wish to see the geometry that resides on the other layers in the model. Separate individual layers and layer lists by a comma. The default range is 0-254, which is all layers.
- View and Position - Allows you to see and place the associated model instance(s).
- Family Member - Toggles the use of family of parts functionality on and off. If enabled, the Member option is unhazed and available. See "Adding a Family Member of a Master Model" on page 3-15.

Please note: Family of Parts functionality is described in the *Parametric Modeling User Guide and Menu Reference*.

- Member - Allows you to specify the name of a Member found in the Table File associated with the Master model. You can select a member of the Master model (model name specified using the Model option) from the optional Family Member List. The Member option is only valid when adding a new component class.

  See "Adding a Family Member of a Master Model" on page 3-15 for more information.

Viewing and Positioning the Component

When you create a component, you can position it in the assembly with or without orienting it. The View and Position options are also available on the Copy Component and Add Assembly property sheets, shown in Chapter 7, "Changing the Assembly Structure".

See "Position and Orient" on page 3-14 for more information about positioning component instances.

After filling in the Add Component property sheet, choose one of the procedures given below.

View and Position without Orienting

1. Choose the View and Position option.

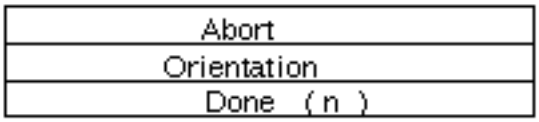
   The default orientation of the model graphics being viewed is determined by the Model Space Origin (MSO), or TOP Cplane of the model associated with the component. The Cplane of the model is planar (or parallel) to the TOP Cplane of the parent component. The default location may cause the model graphics to appear out of position in the assembly. Using View and Position, override the default orientation so that MSO is based on the Active Cplane of the Active Adrawing, instead of the default TOP Cplane.

2. Type a value in the Layer field to view the model graphics on a layer other than the layer on which the model graphics reside in the actual CADDS model.

   The number you specify is added to the layer number on which the actual model graphics resides. The model graphics are then viewed on the layer which is the sum of these two numbers. For example, if the model graphics reside on layer 10, and you want to view them on layer 15, you would enter layer 5 in the layer field.

   If you do not specify a layer, the model graphics appear on the same layer on which they reside in the actual CADDS model, regardless of the active layer of the Adrawing in which you are currently working.

3. Click Apply. A pulldown menu appears with optional Orientation selections, and a Done (n) field, which displays the number of instances of the component to be positioned.



4. To position the component, select a CADDS Location icon (Parametric) or Entity Placement icon (Explicit) from the left bar. You can position it as many times as you want. The last location is the actual position.

5. The component is added to the assembly structure and viewed in the graphics window. Click Done.

   For adding multiple instances, the Done(n) field counts down, indicating the number of instances remaining for you to position. To position the next instance, repeat Step 4.

6. Click Apply. The assembly structure is updated.

Position and Orient

Position the component, then rotate it (any number of times). Now you can locate the component.

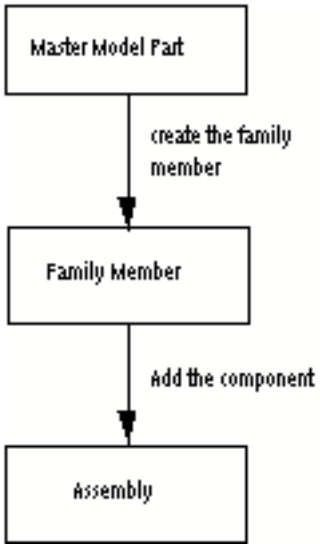
1. Choose the View and Position option from the Add Component property sheet.
2. Enter a value in the Layer field, if desired.
3. Click Apply. A pulldown menu appears. The bottom selection is Done (n)field, which displays the number of instances of the component to be positioned.

   See “Position Component” on page 8-30 for more information on positioning a component instance.

Adding a Family Member of a Master Model

The Family Member and Member options on the Add Component property sheet enable you to associate the new component class to a Family Member of the named Master Model. The master model name must be specified in the Model name field of the Add Component property sheet. The Member name specified must already exist in the Table File of the Master Model.

The regeneration of the Family Member and/or its association to the named component class is performed when you add the component to the active CAMU structure. Like the GENERATE FAMILY command, the Family Member is regenerated from its corresponding Master Model and Table File.



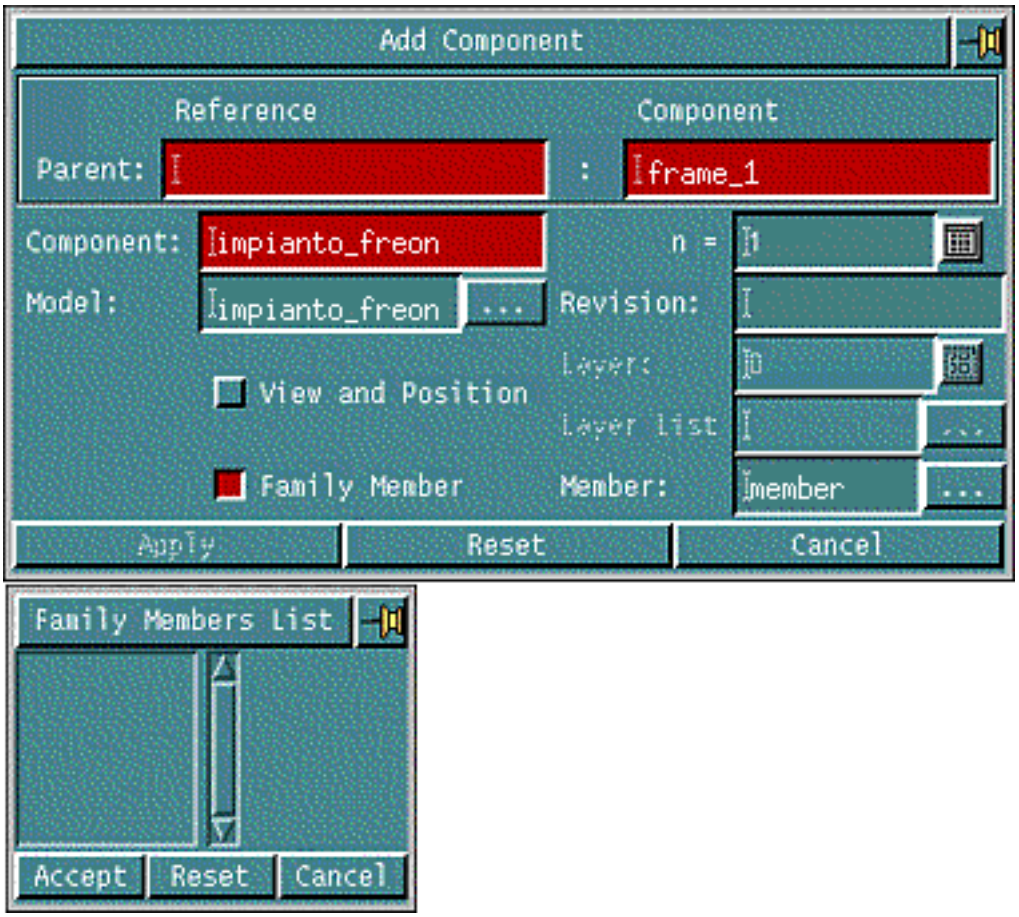
The Member menu option is valid only when a new class of component (new Component Name) is added to the assembly. Then the model corresponding to the selected Family Member is associated with the component. If the associated model is not found, regeneration of the Family Member is performed. The name of the associated model is wholly based on the naming conventions established in the Table File of the master Model for Family Member information.

Please note: The terms Master Model, Table File, and Family Member, as well as Family of Parts functionality in its entirety, are documented in the *Parametric Modeling User Guide and Menu Reference* and in parametric online command help for the GENERATE FAMILY command.

In the following example, the model associated with the Family Member `member1` of the Master Model `impianto-freon` is associated with the new component class `impianto-freon`.

Please note: The Component class name and the Model name can be identical but they do not need to be.

Figure 3-4 Add Component property sheet with Family Member details

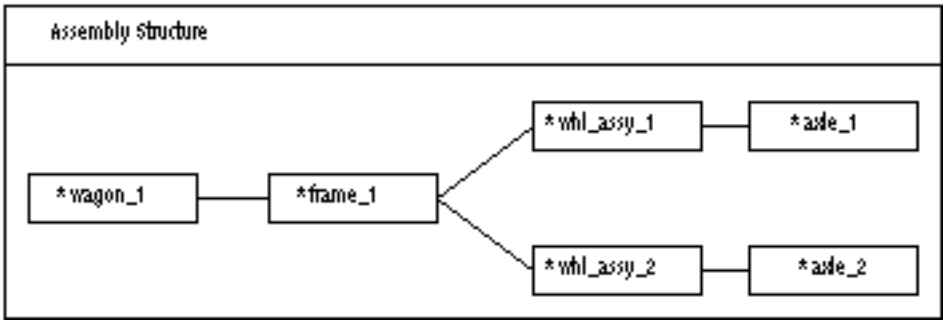


- Model Name: Specifies the name of the Master Model.
- Member Name: Specifies the name of a member found in the Table File associated with the Master Model.

Multiple Occurrences of the Same Item

The same subassembly or piece part can appear in your assembly any number of times. The assembly structure shows all instances of the piece part or subassembly. The components representing these instances have the same class name, followed by an underscore and number representing the instance. For example, the figure below shows an assembly named `wagon_1` that has two wheel subassemblies attached to the frame, each of which have a component `axle_1` and component `axle_2`.

Figure 3-5 Assembly `wagon_1` with two wheel subassemblies



If a model is associated with the component, you can look at the graphic representation of the model in the CADDS Graphics Window. Just as the structure window shows two instances of the class `axle`, the CADDS Graphics window shows two graphic representations of axle. Each instance of the `wheel` subassembly can have its own unique location and orientation in the assembly, and its own associated attributes (user-defined non-graphic information). Initially, all instances of the model are viewed at the same place.

A Parent Component Having Multiple Instances

If more than one instance of the specified parent component exists, the component you create is attached to every instance of the parent. The system assigns instance numbers to the new child (or leaf) components.

Adding Multiple Instances of the Same Component

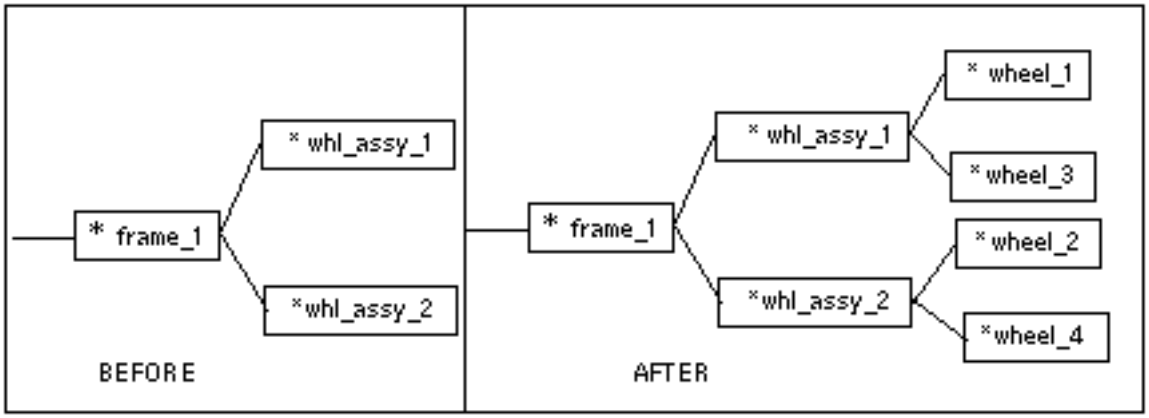
You can add one or more instances of a component class at a time. Choose the Add Component option, or choose Add Component from the popup menu in the Assembly Structure window. Indicate how many instances of that component you want on the property sheet that appears. The default is one (1) instance. If the parent you specify has more than one instance, multiple instances of the new component are attached to every instance of the parent.

In the example below, the user adds two instances of the wheel component to the parent `whl_assy_1`. Because Add Component operates on all instances of the parent component, two instances of wheel are added to each instance of `whl_assy_1`.

Add Component Property Sheet Options Specified

- Specify `whl_assy_1` in the Parent field.
- Specify `wheel` in the Component field.
- Specify a value of `2` in the n (number of instances) field.
- Select the Apply option to process the command.

Figure 3-6 Two Instances added to the parent whl_assy_1



Instance Names

The instance identifier, which comes after the class name and underscore (_) in the component, is unique for each instance. The system assigns numbers as instance identifiers, beginning with 1 and incrementing by 1 for each subsequent instance of the component. You can change the instance identifier to be more product-specific. For example, change `whl_assy_1` and `whl_assy_2` to `axle_front` and `axle_back`. Use Change Component to change a component's instance name. This procedure is described in Chapter 7, "Changing the Assembly Structure".

If you change the class name of a component instance, it no longer has a clone-type relationship with the remaining component instances of that class. You cannot reestablish this relationship after it is removed.

Making a Change to Only One Instance of a Component

Some commands can be chosen to operate on only one instance of a component or all instances. Property sheets for these commands contain an All Instances toggle. Other commands operate on all instances of the component you name (for example, ADD COMPONENT).

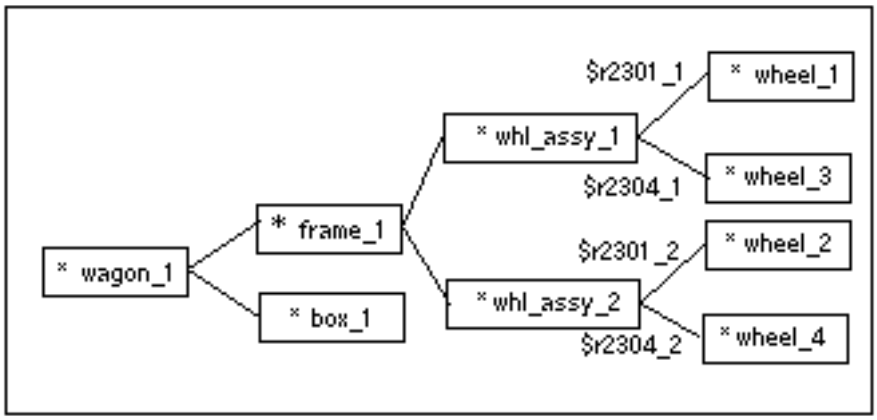
Relations Between Components in the Structure

The relationship between two components is represented by a line, called a relation. Each relation has a self generated name and number. The first part of the name represents the graphic orientation between the two components. The underscore and number indicates how many instances of that relation exist.

Relation names are not shown in the assembly structure. You can select the relation and/or execute the command VERIFY OBJECT to show the relation name in the Explicit environment Text window or Parametric environment Report window.

In the figure below, the relation instance between `whl_assy_1` and `wheel_1` is `$r2301_1`. The relation between `whl_assy_2` and `wheel_3` is `$r2301_2`. Both relation instances have the same relation class name `$r2301`. `whl_assy_1` and `wheel_1` are related to each other in the same way as `whl_assy_2` and `wheel_3`.

Figure 3-7 `wagon` **Assembly Showing Relation Instances**



Component names can contain any combination of alphanumeric characters. In addition, component names can contain a period (.) and a dollar sign ($).

If you plan to use the interpart constraints capabilities of CAMU, see Chapter 9, “Using Interpart Constraints”.

Adding Existing Assemblies to Your Assembly

There are two ways to add an existing, external assembly to the assembly you are designing.

- Add a reference to the external assembly

  When you add an external assembly to your assembly, a component (assembly instance) representing the root component of the external assembly is added to the parent node you specify in the active assembly.

- Add a copy of the external assembly

  The copied assembly becomes an integral part of your active assembly. It is no longer associated with the original assembly database from which it was copied.

Each method adds a component (representing the root node of the external assembly) to a parent node you specify in the active assembly. The differences between adding an assembly as a "reference" and adding an assembly as a "copy" are explained in "Reference Assemblies" and "Copied Assemblies" on page 3-22.

Reference Assemblies

Adding the external assembly as a reference assembly traditionally gives you read-only access to it. You can view it within your active main assembly, change its orientation and location (as a unit in model space), and change its parent relation. However, you can lock any component within it, change its structure and even activate and change any model associated with its leaf components.

You can make changes to the reference assembly, provided that you can obtain a lock on the desired components. For example, you can activate the model associated with a component class resident in a reference assembly - provided that you can first perform a Lock Component operation on the component instance.

Please note: You can activate and edit a reference assembly from in the active main assembly by locking a child component of the associated assembly instance. The act of locking any child of a reference assembly instance activates the associated assembly database (`_db`).

To activate a model in a reference assembly, the component associated with the model must be locked. The newly activated model will inherit the viewstate from the suspended Adrawing or model.

A reference assembly is a reference to another assembly that exists outside of your active main assembly. You can see any changes made to that original assembly by refreshing or filing your active main assembly. Any time you file or refresh the assembly, any changes to the reference assembly are reflected in Assembly structure as well as the Graphic window, if any models associated with the referenced assembly components are changed.

Use CHANGE ASSEMBLY to assign the reference assembly to a new parent or to change the reference assembly's name in the active assembly. For more information, see Chapter 7, "Changing the Assembly Structure".

The root node of a referenced assembly is called an Assembly Instance. An assembly instance is shown in a bold-outlined box in the assembly structure. This bold representation is only used to indicate an assembly instance.

You can choose to display the entire reference assembly or only its root component. You can display any part of the subassembly using the Display Structure property sheet.

Please note: You cannot move piece parts that belong to reference assemblies except to change the appearance within exploded views. See "The Exploded View" on page 10-10.

Copied Assemblies

You can add a copy of an external assembly to your active assembly. The copied assembly becomes an integral part of your active assembly. When you add a copy of an external assembly to your active assembly, it is no longer associated with the original assembly database from which it was copied. This means that you do not see any changes made to the original assembly.

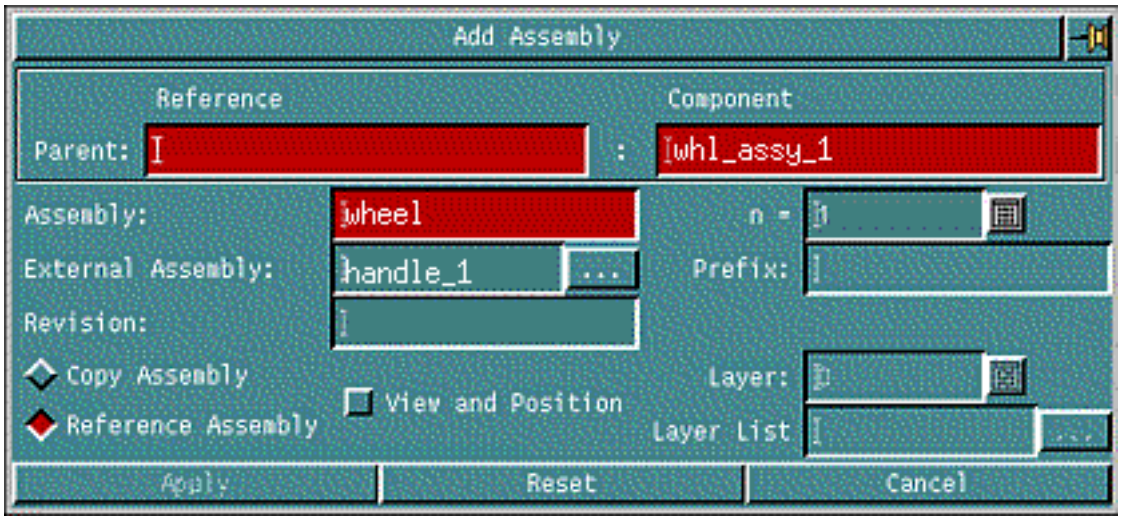
You can manipulate and orient the root component, change the assembly structure, attach non-graphic information (attributes) to it, manipulate and change the model geometry associated with leaf components, providing the appropriate components are locked.

Use CHANGE COMPONENT to change a component name, assign a new parent, or change an instance name for components of assemblies you have added (copied). For more information see Chapter 7, "Changing the Assembly Structure".

Adding a Reference Assembly

Use Add Assembly to add a reference to an existing, external assembly to your active assembly. Perform the following steps.

1. Choose the Add Assembly option from the COMP/ASSY task set. The Add Assembly property sheet appears.



   The component selected in the assembly structure or the graphics window appears in the Parent field.

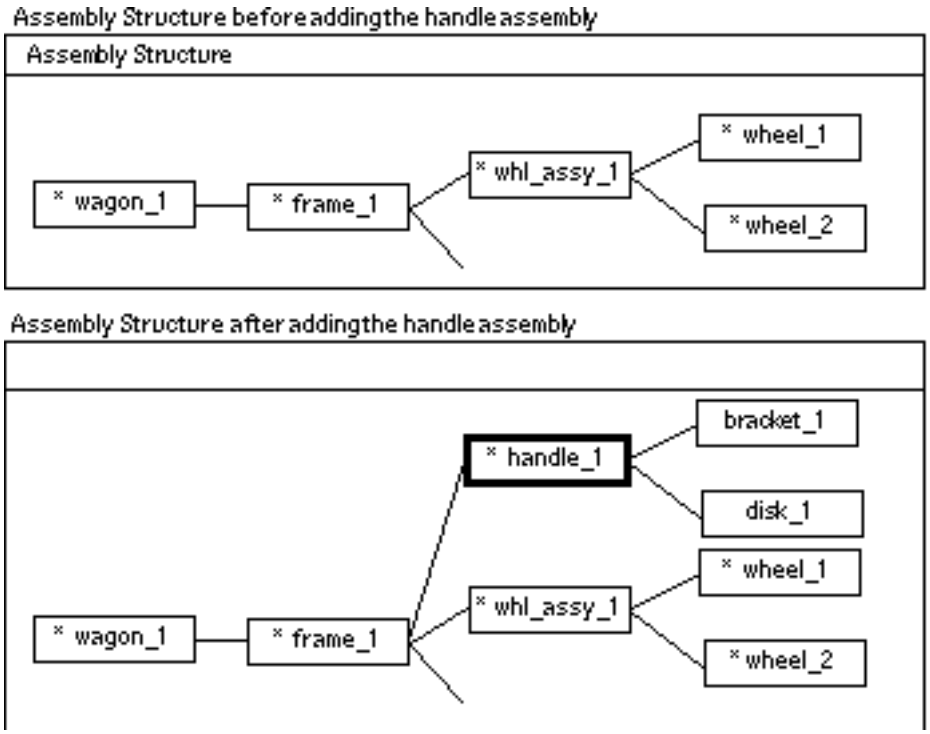
2. Select the parent component. Enter the parent name in the Parent field. In this example, `whl_assy_1` is the parent name.
3. Enter the name of the root component in the Assembly field. In this case it is wheel.
4. Enter the name of the external assembly (as it exists in your create directory), in the External Assembly field.
5. The Copy Assembly option on the property sheet is selected by default.
6. Click Reference Assembly. You can view and position the reference or inserted assembly within your active assembly.
7. Enter a prefix to resolve name conflicts.

   If the assembly you are adding (copying) has a component with the same name as an existing component in your active assembly, supply a prefix and an optional delimiter. The same applies for component instance, associated attribute group name, or attribute name. The prefix and delimiter are added to the front of the component name of the external assembly that you are adding to the active assembly. The prefix, plus a delimiter, has a maximum of seven characters.

Example of Referencing an Assembly

In this example, an existing assembly named `handle` is referenced into the active assembly `wagon`. As a result the `handle` subassembly's root component appears in a bold box.

Figure 3-8 Using the `handle` Assembly As a Reference Assembly in the `wagon` Assembly



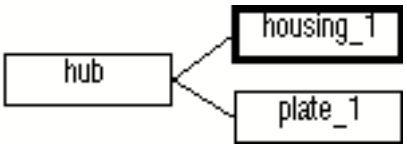
Nested Reference Assemblies

When you add a reference assembly to your active assembly, the reference assembly too can have a reference assembly(s) within it. This is called a nested reference assembly. In that case, CAMU assigns a dot extension to the nested assembly name. This preserves the instance portion of the instance name of the nested reference assembly. You cannot change the dot extension.

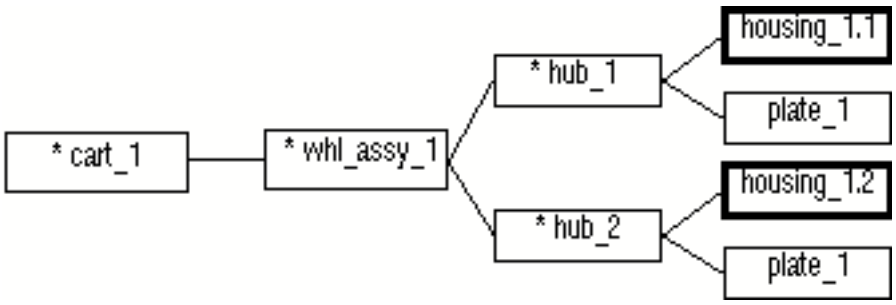
In your active assembly, the parent components of both of the reference assembly and the nested reference assembly appear in a bold box.

Example of a Nested Reference Assembly

The figure below shows the assembly structure for the `hub. It` contains a reference assembly (`housing_1`) instance. The `hub` assembly will be added to another assembly as a reference.



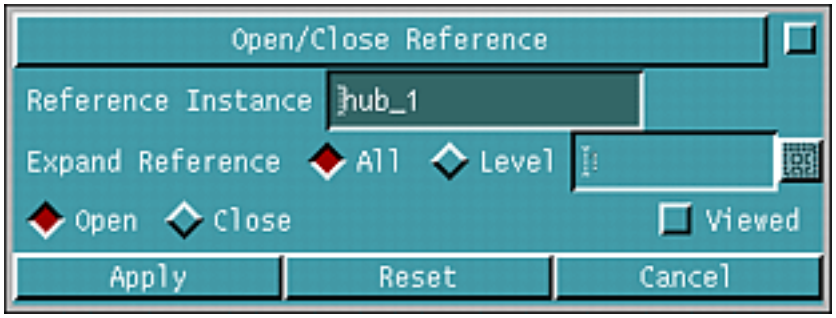
When you activate an assembly named `cart`, and add two instances of the `hub` assembly to it.



The instances of `hub` appear in bold because they are reference assemblies. The instances of `housing`, the nested reference assemblies, also appear in bold. In addition, each instance of housing has been assigned a dot extension.

Controlling the Level of Nested Reference Assemblies

Use the OPEN/CLOSE REFERENCE option to read and display the desired level of reference assemblies of the active main assembly in the assembly tree window. Select the COMP/ASSY menu from the top bar to access this option. The Open/Close Reference property sheet appears as shown:



- Reference Instance: The reference instance name that you want to open or close.
- Expand Reference: The level up to which you can specify to read and display the reference assembly in the assembly structure tree.
  - All: Read and display all the reference assemblies.
  - Level: Read and display the specified level of reference assemblies. You can specify any real number greater than zero. The default number is 1.
- Open: Read and display the assembly structure of only the open reference assemblies. This is set on by default.
- Close: Curtail the assembly structure up to the specified reference assembly node. The specified reference assembly node is marked with >> symbol in the assembly structure.

Please note: You cannot close an active reference assembly.

- Viewed: Display the latest saved viewstate of the components in the open assemblies. Use this option with Open option only.

Please note: To save the current view states of a reference assembly, the reference assembly must be open while saving the main assembly. The view states are lost when the reference assembly is closed. For more information about the view states of a reference assembly, refer to TAN number 123167.

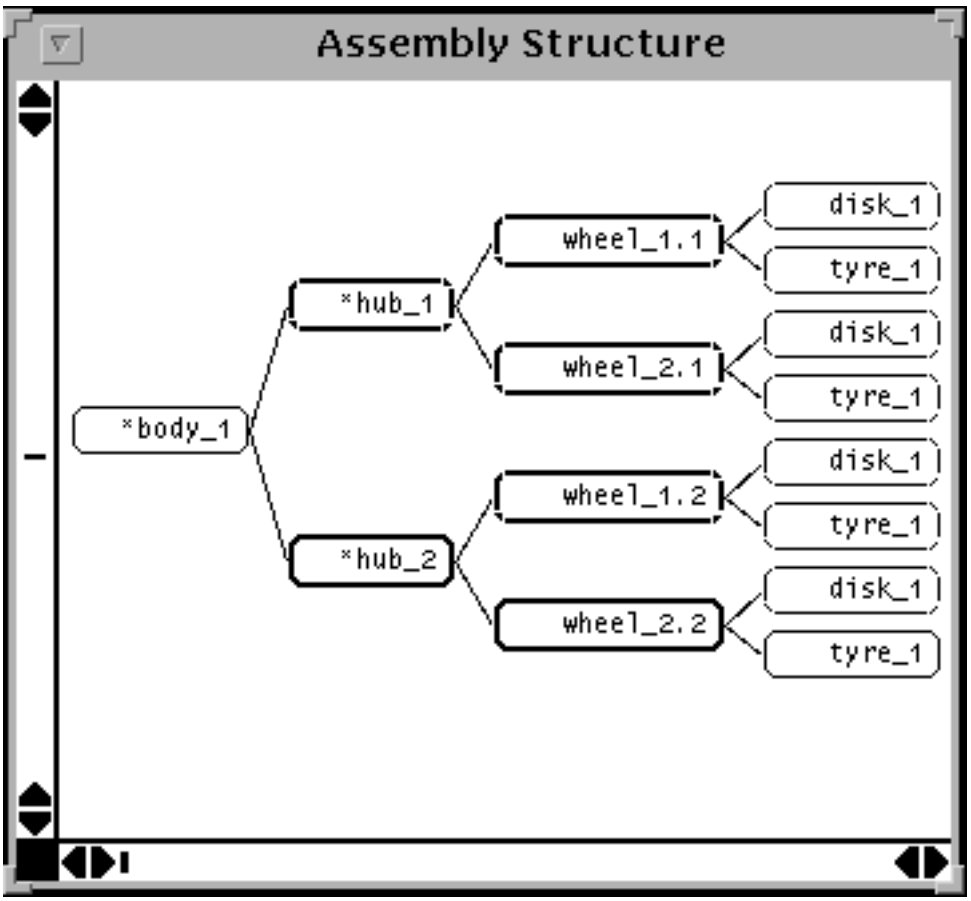
A >> symbol on a reference assembly node in the assembly structure tree, denotes that it contains further more reference assemblies in close state.

You can also access the Open Reference and Close Reference options from the context sensitive menu within the assembly structure window.

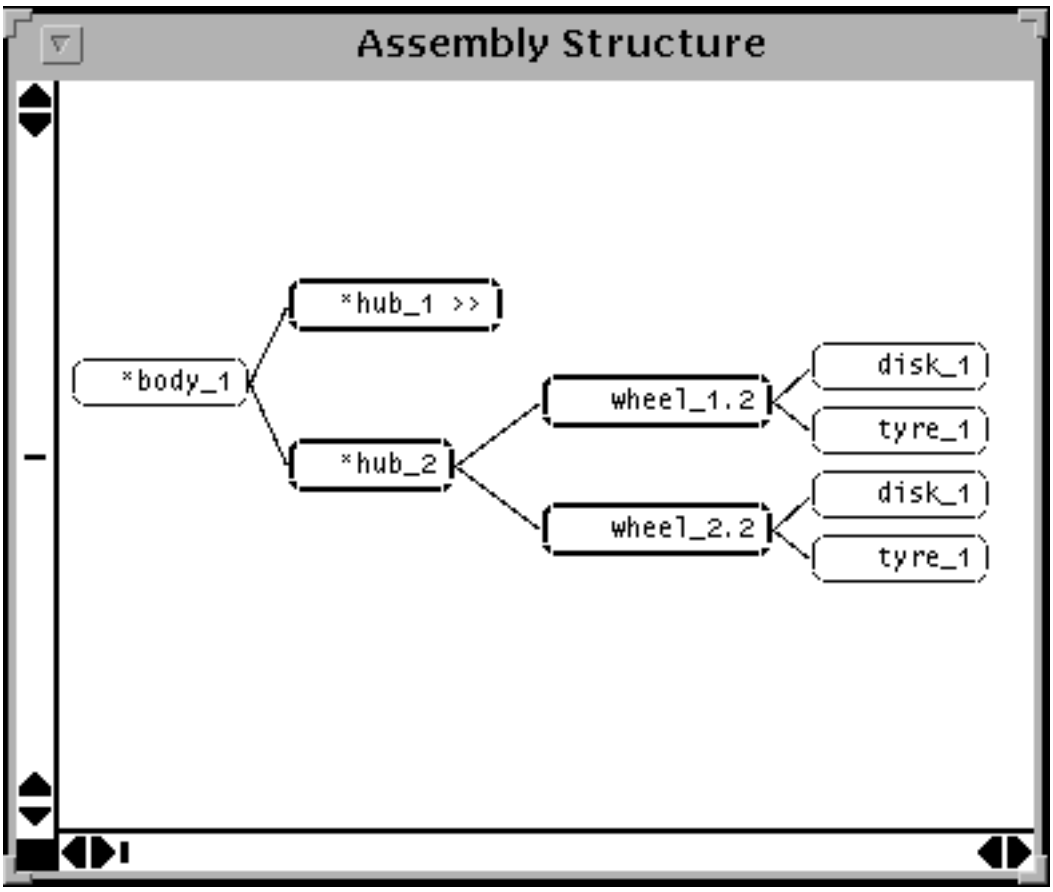
Please note: You cannot export subassemblies having a closed reference assembly in any of its instances.

Example: Open/Close Reference

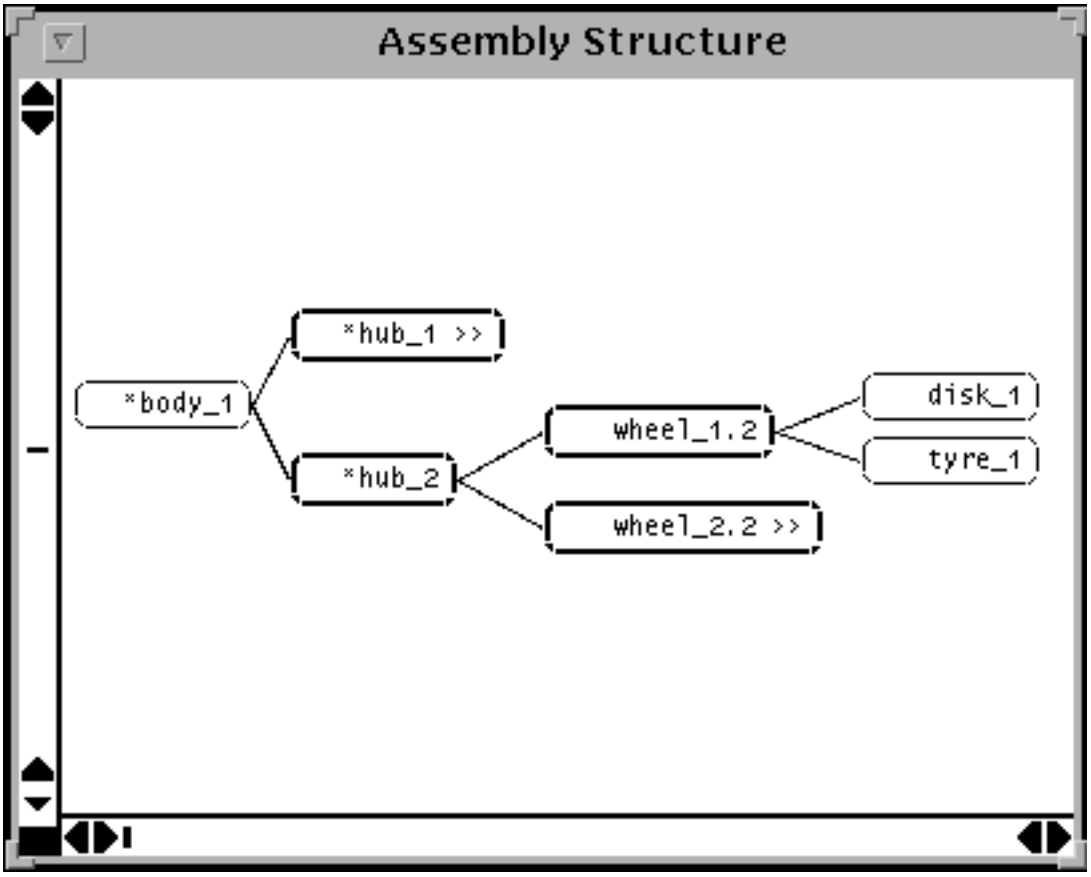
Following is an assembly structure tree of an open assembly `body_1`. It has two instances of reference assembly hub, as `hub_1` and `hub_2`. Each of these hub assemblies further have two instances of reference assembly wheel, as `wheel_1` and `wheel_2`.



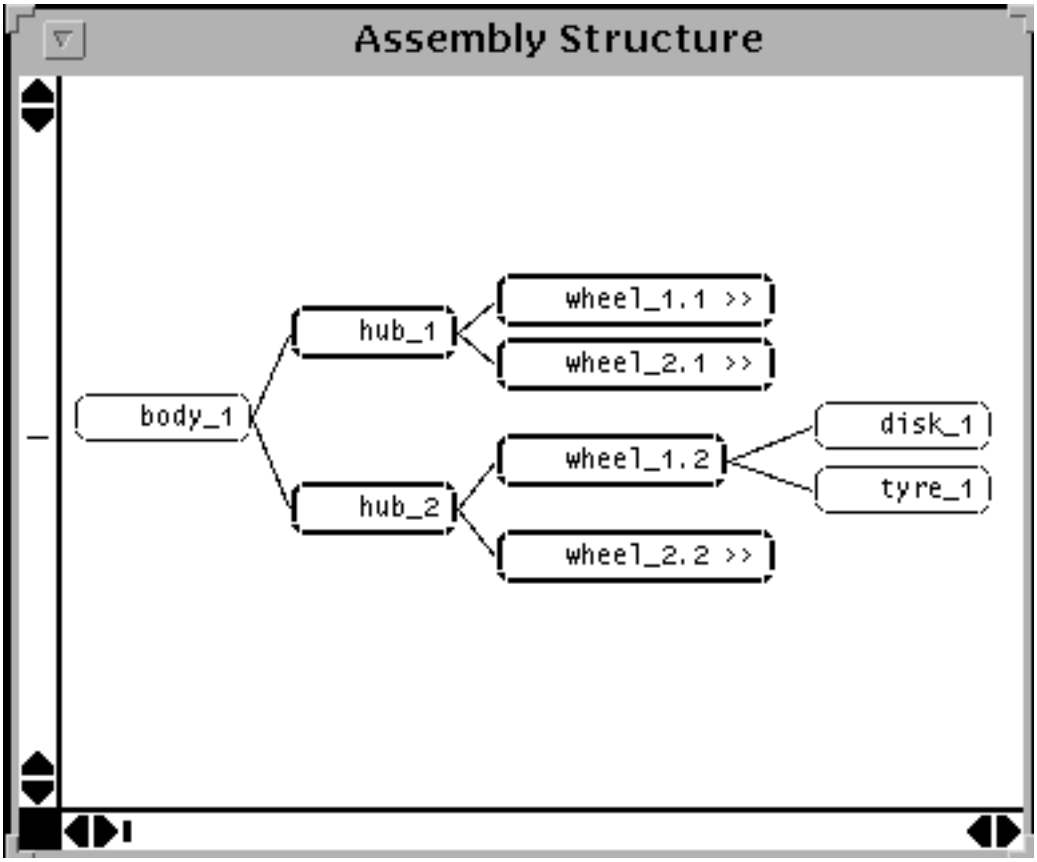
Select the reference instance `hub_1` and choose the Close Reference option on the Open/Close Reference property sheet. The >> symbol appears on the `hub_1` reference assembly node, curtailing the rest of the reference assemblies, as shown in the given figure.



Select the reference instance `wheel_2.2`, and choose the Close reference option on the Open/Close Reference property sheet. The >> symbol appears on the `wheel_2.2` reference assembly node, curtailing the rest of the reference assemblies, as shown in the given figure.



Select the reference instance hub_1, and choose the Open Reference option on the Open/Close Reference property sheet assigning a value of 1 for the Level option. This opens the reference assembly hub_1 up to one level, that is, the two instances of reference assembly wheel are displayed, as shown in the given figure.



Using the Prefix Option

You can use the prefix option to maintain class name uniqueness in the event that the assembly you are adding contains class names which are already present in the active assembly. For example, the active assembly contains a class named `disk`. The existing assembly that you are adding also contains the component class `disk`. Use the Prefix option, and a value such as `hd`, to prefix the incoming instances of class `disk` as `hd-disk`.

Creating a New Assembly by Exporting

You can create an assembly database from any subassembly in the active main assembly, including a reference assembly. You can also transfer Adrawings from the main assembly to the exported assembly. These Adrawings contain view states associated with the assembly selected for exporting. You can optionally import the selected subassembly as a reference assembly in the active main assembly.

You can file the active assembly, or any reference assembly within it, with a new name by exporting the root component of the active main assembly (or the assembly instance of a reference assembly).

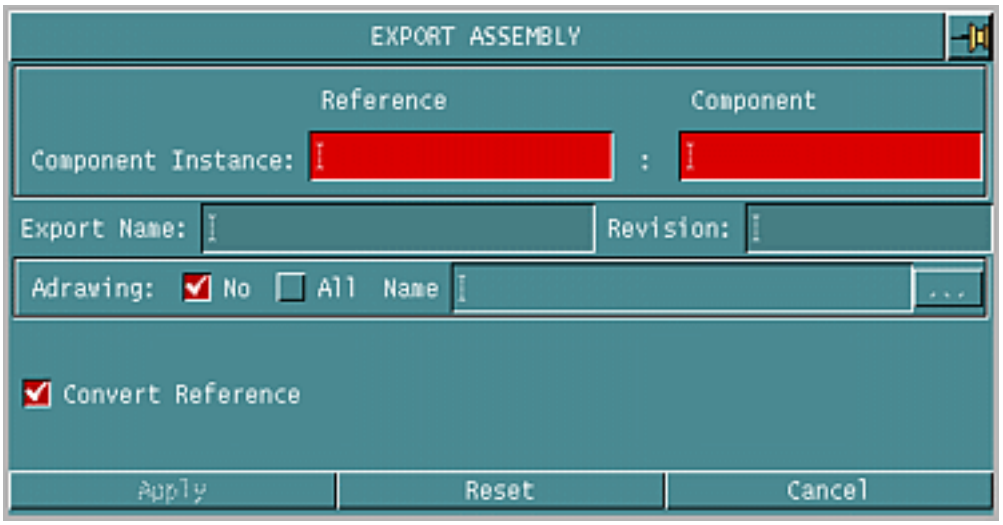
You can maintain different versions of the same assembly by saving the assembly with different revisions.

To export an assembly, proceed as follows:

1. Choose the Export Assembly option from the COMP/ASSY task set.

The Export Assembly property sheet appears.

Figure 3-9 Export Assembly Property Sheet



2. Choose the Component from the active or reference assembly to be exported as a new assembly. (Or enter the component name in the Component Instance field.)
3. Enter the name for the exported assembly in the Export Name field.
4. Specify the revision of the assembly in the Revision field, if the same assembly is to be saved with a different revision.
5. Specify the Adrawings associated with the selected assembly that you want to copy to the exported assembly. Choose one of the following options to select the Adrawings:
   - No: To not copy any of the Adrawings to the exported assembly. If you do not select an Adrawing, the No option is checked by default.
   - All: To copy all Adrawings to the exported assembly.
   - Name: To copy only selected Adrawings to the exported assembly. You can specify the name of the Adrawing in the Name field or select the Adrawings from the runtime list of Adrawings associated with the selected assembly.
6. Choose Convert Reference option if you want to convert the selected subassembly to a reference assembly.

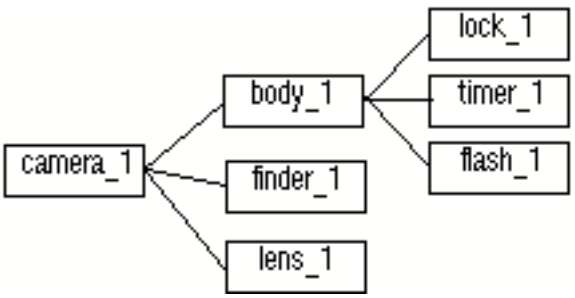
You can export the view states from the active and the filed Adrawings, but not from a suspended Adrawing.

You cannot export the Adrawing of an assembly if you do not have necessary permissions to read the Adrawing parts of the main assembly.

Example: Exporting without Referencing

In this example, create a new assembly using a portion of the active main assembly. Do not specify the selected subassembly as a reference assembly.

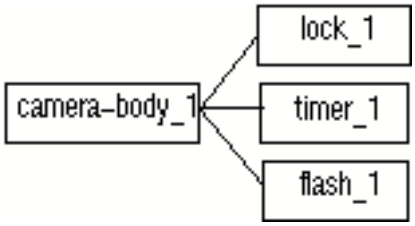
1. Activate the assembly that contains the subassembly you want to export. In this example, the active main assembly is `camera`.



2. Select the subassembly component `body_1` to export.
3. Open the Export Assembly property sheet.

   The selected component instance name appears in the Component field.
4. Specify a new assembly name `camera-body` in the Export Name field.
5. Do not enable the Convert Reference option.
6. Apply the property sheet.

The following new assembly is created.



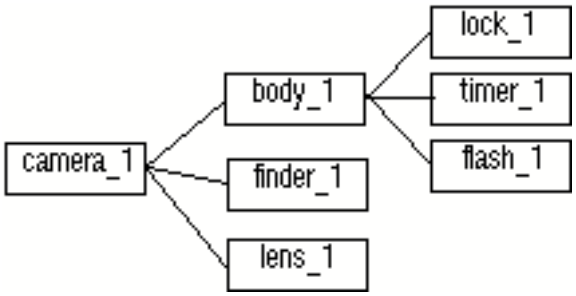
The active main assembly remains unchanged.

Example: Exporting with Referencing

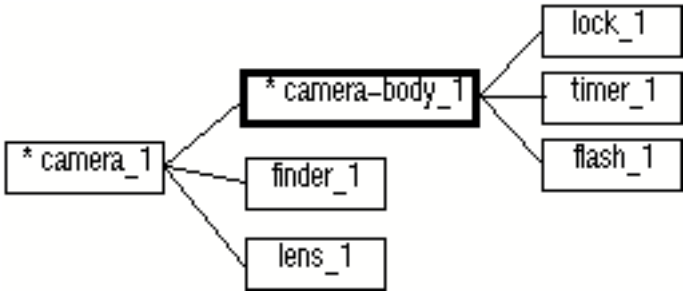
In this example, create a new assembly using a portion of the active main assembly. Automatically delete and then import the selected subassembly as a reference assembly.

All components in the source subassembly, `body_1` and its children, and the parent subassembly node, `camera_1`, must be locked.

Please note: You cannot select and export a subassembly whose top level component is associated to a model. For example, if you select `body_1` in the structure below, and `body` is associated to a model, the EXPORT ASSEMBLY command would fail. The root node is the only exception to this rule. You can select the root node (`camera_1`) and use the EXPORT ASSEMBLY command to create a duplicate assembly structure with a new name. See “Exporting the Active Assembly” on page 3-35.

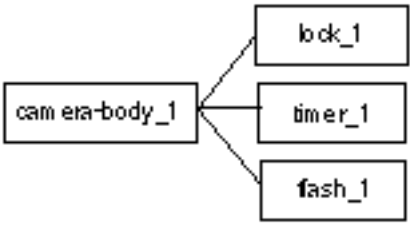


Repeat the procedure in "Example: Exporting without Referencing" on page 3-32, but enable the Convert Reference option on the Export Assembly property sheet. The resulting assembly structure will be:



The following assembly is created.

Please note: All associations in the original assembly are maintained in the converted reference except parametric positional constraints. If you had used a command such as DISPLAY STRUCTURE COMPRESS, the structure display is duplicated in the reference.

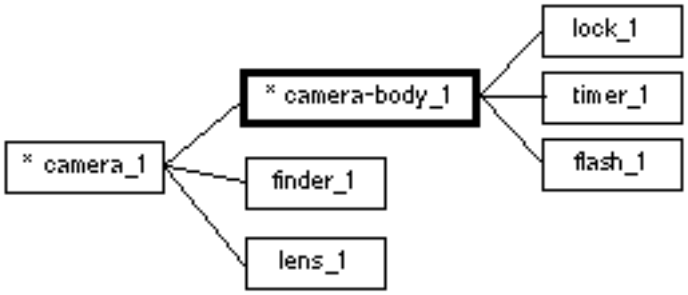


The sub-assembly `bxdy_1` is converted to reference assembly `camera-body`. The `camera-body/_db` file is created. No Adrawings are copied or created for the new assembly.

Exporting a Reference Assembly

You can create a duplicate of a reference assembly as a new name from the active main or reference assembly.

1. Activate an assembly. In this example, the active main assembly is `camera/_db` and the active Adrawing is `default`.



2. Select the assembly instance (root component of the desired reference assembly) to export. In this example, select `camera-body_1`.
3. Open the Export Assembly property sheet.

   The selected assembly instance name appears in the Reference field.
4. Specify a new assembly name in the Export Name field. In this example, the new assembly name is `camera-body-v2`.
5. Do not enable the Convert Reference option.

Please note: The Convert Reference option is ignored when the source component is a reference assembly instance.

6. Apply the property sheet.

A new assembly `camera-body-v2` is created. It is exactly the same as the `camera-body` assembly.

Please note: See also “Example: Exporting with Referencing” on page 3-33.

Exporting the Active Assembly

You can create a new assembly database that is identical to the active main assembly. This essentially enables you to file the active main assembly as a new name.

Chapter 4

Filing an Assembly

This chapter describes the different options available for filing your assembly. The topics are:

- Filing Your Changes
- Filing or Quitting Selected Adrawings or Models

Filing Your Changes

While working on the Assembly structure or model, periodically file your work. When you do so any changes made during that work session are written to the assembly database. You can file an entire assembly or selected Adrawings and/or models.

The default Adrawing and root node model are filed whenever you file the assembly structure. When you file suspended Adrawings or models, the system deactivates the suspension and updates the database with all the changes made before it was suspended.

If other users are also working on the active main assembly, filing commits your own changes to the database, then refreshes the assembly to show you any changes that other users have filed since you began working in the assembly (or since you last filed, whichever was most recent). After you file, other users can update their copy of the assembly with your changes by filing or refreshing the assembly from their own system. For more information about filing and refreshing assemblies in a multi-user environment, see Chapter 5, “Working in an Existing Assembly”.

If your active main assembly includes a reference assembly, the reference assembly is also updated; that is, it reflects any changes that have been made and filed to its associated external assembly since you last filed your active assembly.

When you file the assembly, all active reference assemblies are also filed. See “Updating Active Reference Assemblies” on page 4-3.

Positional constraints are saved when you file.

Please note: If you have issued Select Assemble On, see “Automatic Assembly” on page 6-32, all parametric positioning constraints are solved when you file.

Updating Associative Annotation

Filing the assembly updates all associative dimensions, HLR, hatch, and cross sections.

Updating Active Reference Assemblies

If you have activated and edited a reference assembly within the active main assembly, filing the active main assembly commits those changes. Once committed, you can optionally deactivate active reference assemblies using the QUIT REFERENCE command.

Deactivating a Reference Assembly

QUIT REFERENCE enables you to de-activate an active reference assembly. The assembly becomes read only. This allows you to activate another reference assembly when the maximum number of active reference assemblies is reached.

Please note: Before you can issue this command, you must file the active main assembly. This commits the changes you have made to the active reference assembly(s).

1. Choose File -> Quit Reference from the top bar.

   This opens the Quit Reference property sheet.

2. Specify the desired reference assembly in the Reference field.

Understanding System Messages

If you attempt to deactivate an inactive reference assembly, following message appears:

```
ERROR: You can't deactivate an inactive assembly.
```

When the assembly has modifications that have not yet been committed, following message appears:

```
ERROR: The assembly%name% still has changes that have not been
committed yet. Please use the FILE ASSEMBLY command to file the
changes.
```

Filing All Active Assemblies

The FILE ASSEMBLY command files the active main assembly and all active reference assemblies.

Filing and Locking Interactivity

Regular locks (LOCK COMPONENT) remain when you file the assembly. See "Locking Areas of the Structure" on page 5-32.

Filing Assembly Structure, Default Adrawing, and Root Node Model

The active main assembly, all active reference assemblies, the `default` Adrawing, and the `root` node model are filed with the FILE ASSEMBLY option.

Click FILE ASSEMBLY, or click FILE ASSEMBLY W/ OPTIONS and then select the Assembly Structure option on the property sheet. With this the structures, `root` node model, and the `default` Adrawing are updated.

Filing with and without Exiting

If you file and exit, you return to the LDM menu. If you file an assembly and all suspended items, but do not exit, both the active part (model or Adrawing) and all suspended Adrawings and/or models are updated with your changes. You remain where you were before you filed the assembly.

Selections from the FILE Icon

When you select FILE from the top bar, the following menu appears:

FILE ASSEMBLY
FILE ASSEMBLY W/ OPTIONS...
FILE AND EXIT
FILE AND EXIT W/ OPTIONS...
QUIT SUSPENDED...
QUIT REFERENCE...
QUIT AND EXIT
CONSTRUCT PART
INSERT PART
EXIT CADDS 5

Table 4-1 File menu options

Menu Option	What Happens
FILE ASSEMBLY	All changes made to any Adrawings, models, or assembly structures in your active assembly during this work session are added to the database. You remain in the active assembly and in the active model or Adrawing you were in when you selected FILE ASSEMBLY. FILE ASSEMBLY files the active main assembly and all active reference assemblies.

Table 4-1 File menu options

Menu Option	What Happens
FILE ASSEMBLY W/OPTIONS	A property sheet appears, listing your active and suspended Adrawings and models. To file the assembly structure, default Adrawing, and the root node model, select the Assembly Structure button. For the Adrawings and/or models you select, changes are added to the databases. Models remain viewed. You can file a specified model with a new name.
FILE & EXIT	All the changes are added to the database. Suspended items are unsuspended, then updated. You are returned to the LDM menu.
FILE & EXIT W/ OPTIONS	A property sheet appears, listing your active and/or suspended Adrawings and models. To file the assembly structure, default Adrawing and the root node model, select the Assembly Structure button. To file other items, select from the list of Adrawings or models. To file selected Adrawings and the Assembly Structure, file the Adrawings you select. Apply and file the structure. Apply again.
QUIT SUSPENDED	On the property sheet, select the suspended Adrawing or model you want to quit. Changes since the last filing are not saved in the database, although they are visible on your screen.
QUIT REFERENCE	Enables you to close an open reference assembly. See “Deactivating a Reference Assembly” on page 4-3.
QUIT & EXIT	Any changes made since the last time you filed are not saved. Returns you to the LDM menu.
CONSTRUCT/INSERT PART	These options are available only in the Explicit modeling or Explicit Assembly environments. They enable you to create or insert a CADDS 4X type part. See online command help for more information about these commands.
EXIT CADDS 5	You receive a warning message that you will be returned to system level and if you want to save your changes. Select Yes to save changes and exit the assembly, and return to operating system level. Select No to not file (and thereby lose) your changes, exit the assembly, and return to the operating system level. Select Cancel to stay in the assembly and not file.

Please note: Do not use the EXIT CADDS 5 option while active in an assembly. The session may not terminate completely. Instead, use the File and Exit option. Exit CADDS from the LDM.

Please note: When issuing File or File & Exit Assembly, the system immediately returns a prompt and allows input of commands, even though the application is still under the process of filing and the commands are ignored. This can lead you into believing that the session is hung. Therefore, when filing or exiting an assembly, wait for the message that the assembly and the model parts have been filed before attempting to execute further commands.

Filing or Quitting Selected Adrawings or Models

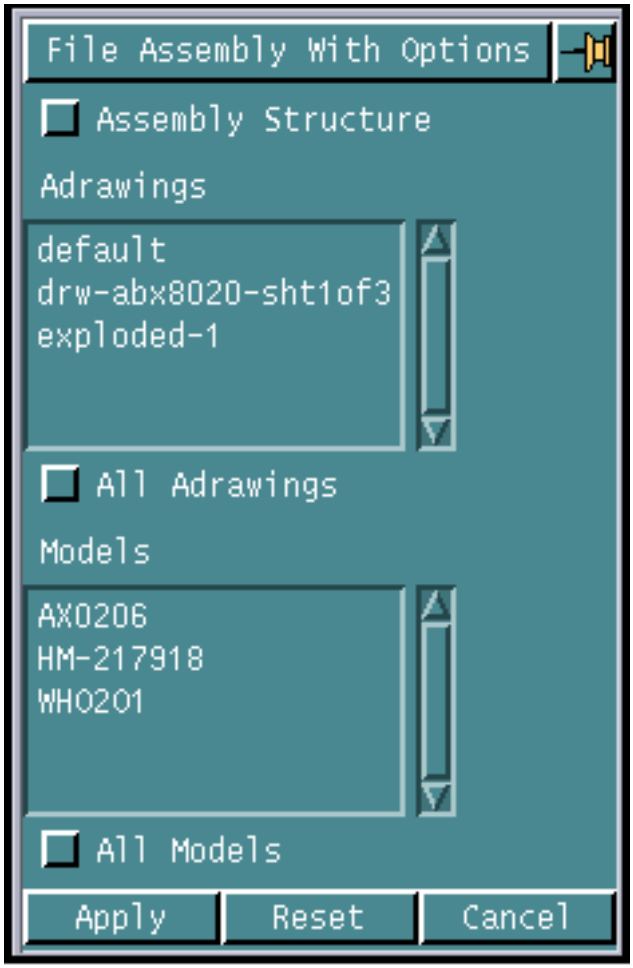
You can specify which items to file (see "Filing with Options" on page 4-6) or quit (see "Quitting with Options" on page 4-8) using options on the FILE menu.

The New Model option enables you to file the specified model as a new name of your choosing.

Filing with Options

Choose FILE -> FILE ASSEMBLY W/OPTIONS to select all or specific Adrawings and/or models to file.

Figure 4-1 File Assembly with Options Property Sheet



1. Choose the Assembly Structure option to file the Assembly Structure (`_db`) itself, the `default` Adrawing, and the `root` node model.
2. Choose the All Adrawings option to file all Adrawings within the assembly that are either active or suspended. You could also file specific Adrawings by selecting their name from the list.
3. Choose the All Models option to file all models that are either active or suspended. You could also file specific models by selecting their name from the list.
4. Click Apply to process your selections.

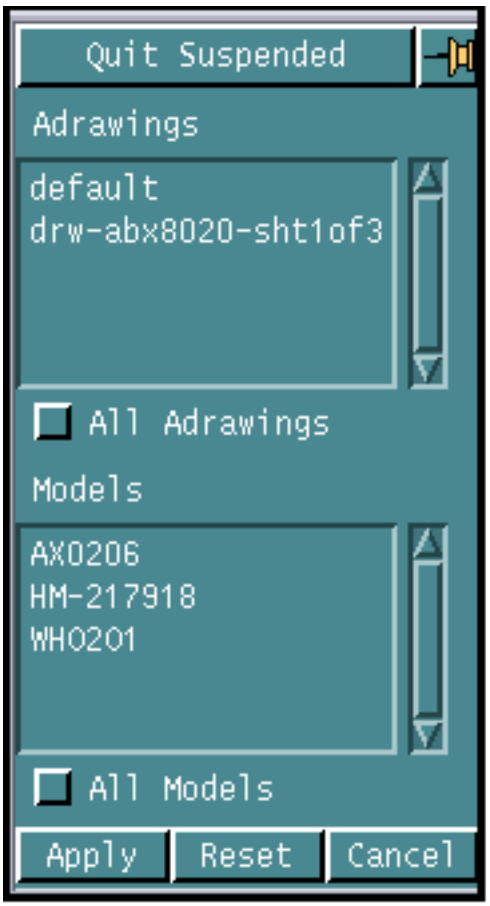
Filing the Active Model Only

CAMU does not have a FILE PART command equivalent. To file the active model only, simply select File Assembly with Options and select only the name of the active model.

Quitting with Options

1. Choose QUIT SUSPENDED to obtain the Quit Suspended property sheet. These options enable you to select all or specific Adrawings and/or models you want to quit. Edits made to the suspended parts that have not yet been filed are discarded when you quit.

Figure 4-2 Quit Suspended Property Sheet



These options allow you to select all or specific Adrawings and/or models you want to suspend. After selecting, click Apply to process the property sheet.

2. Choose the All Adrawings option to quit all suspended Adrawings within the assembly. You could also quit specific Adrawings by selecting their name from the list.
3. Choose the All Models option to quit all suspended models. You could also quit specific models by selecting their name from the list.
4. Click Apply to process your selections.

Chapter 5

Working in an Existing Assembly

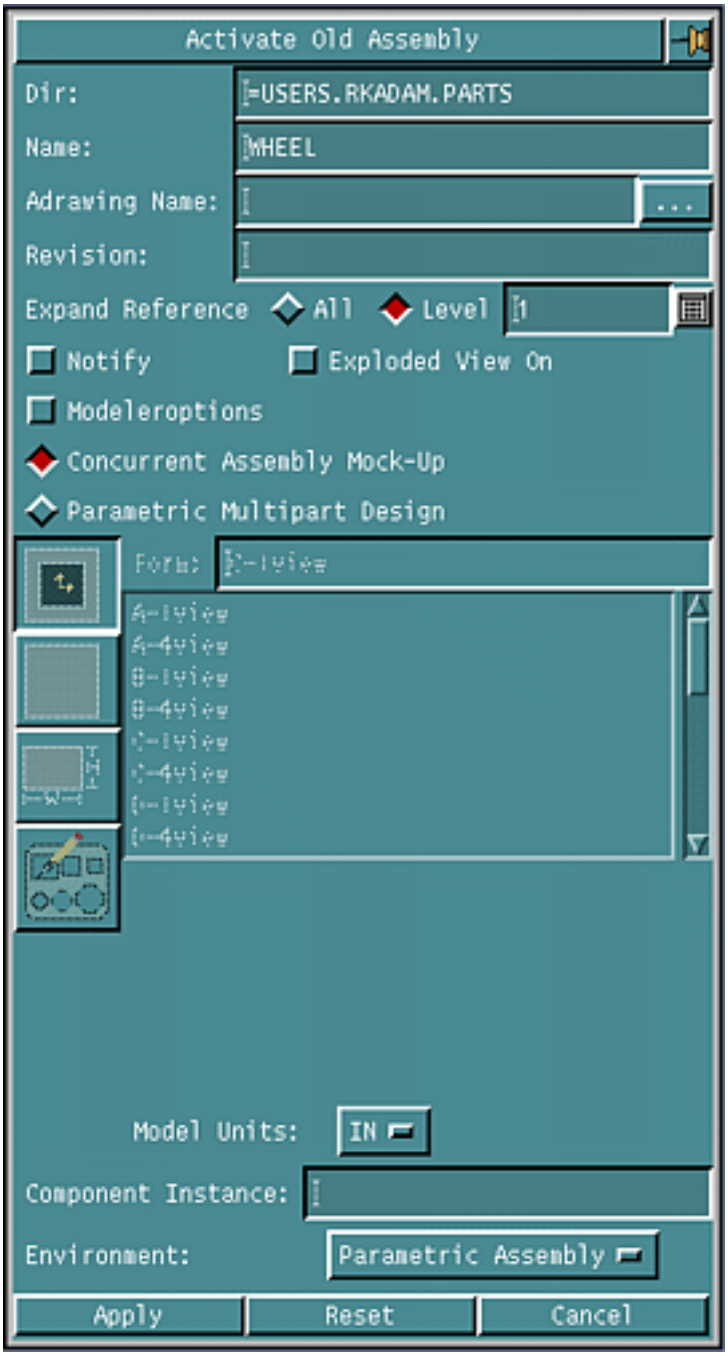
This chapter describes what happens when one or more users work in the same assembly. It also explains, with examples, what happens when you move between Adrawings and models. The topics are:

- Activating the Main Assembly
- Viewing Model Geometry
- Moving Between Models and Adrawings
- Associative Mass Points
- Several Users Can Work Concurrently
- Locking Components, Attribute Groups, and Models
- Activating a Reference Assembly
- Refreshing the Main Assembly

Activating the Main Assembly

You can activate a new or existing assembly from the LDM. This section describes how to activate an existing assembly. From the LDM, select an existing assembly.

Figure 5-1 Activate Assembly property sheet



Please note: You can activate an assembly created in a CADDS version older than the current one. However, you can not activate an assembly created in a CADDS version later that the current one.

Update the necessary information on the Activate Old Assembly property sheet or use the defaults.

1. Click Concurrent Assembly Mock-Up to access the CAMU environment.
2. Enter the name of the Adrawing you wish to activate in the Adrawing Name field. If you do not enter a name, the system activates the `default` Adrawing.
3. The Expand Reference option allows you to specify the level of reference assemblies that you want to read and display in the assembly tree window.
   - Choose the All option to read and display all the reference assemblies.
   - Choose the Level option to read and display the desired level of reference assemblies. You can specify any real number greater than zero. The default number is 1.
4. Choose the Notify option to display a list of models that have been modified more recently than those in the view state of the Adrawing name you specify.

   When you activate an assembly, the following message appears in the Text window (Explicit) or Message window (Parametric):

   ```
   The following models have been modified more recently than
   this Adrawing.
   ```

   Please note: You are notified only about changes to models that were viewed in the Adrawing the last time the Adrawing was filed.
5. Choose the Exploded View On option to have the Adrawing you select appear in its exploded view orientation.
6. Enter the name of a component instance in the Component Instance field. The system activates the model associated with the component. If it is a new model, you can specify modeling units.
7. Click on the list button next to the Environment field to display a list of environments. Choose the environment in which to activate the assembly.

8. Choose the Modeleroptions option to name and activate a new Adrawing. Choose the form, standard sheet, nonstandard sheet, or template icon.

   The icons on the property sheet allow you to select standard sheet sizes with or without predefined views. Choose nonstandard sheet sizes without predefined views, or select from a list of templates.

   Templates save time if you have to create an Adrawing similar to the one that already exists. To activate a new Adrawing and select other modeler options, access the assembly by activating an existing Adrawing. Then activate an Adrawing from the assembly.

9. Click Apply.

While activating an assembly, please note the following:

- You can activate and edit a reference assembly from the active main assembly by locking a child component of the associated assembly instance. The act of locking any child of a reference assembly instance activates the associated assembly database (`_db`).
- You can list all active and non-active reference assemblies (see "Listing Reference Assemblies" on page 7-39).
- The Modify Position Constraints menu may appear when you activate an existing assembly (see "Modify Position Constraints:" on page 6-17).
- To hasten assembly activation time, activate an Adrawing (such as `default` or some other agreed upon name) that contains no viewed components or geometry.

Viewing Model Geometry

When you activate an existing assembly, none of the components or associated models are locked. You can lock components explicitly.

All model instances that were visible the last time the active Adrawing was saved are viewed in the Graphics window.

About Adrawings

An Adrawing provides a workspace on which you can view and manipulate viewed model instances. An Adrawing is both a CADDS part and a drawing. The Adrawing contains views that allows you to see the model space with a particular orientation and scale that you define.

The Purpose of Adrawings

Adrawings allow you to define different views of the entire assembly, subassemblies, or specific components of the assembly. Each assembly can have many Adrawings, each with its own view state. This includes location, number of views, orientation, zoom, scale, and viewed model instances. You can also include annotations within an Adrawing to highlight viewed geometry. You can turn off the view of certain components on one Adrawing and have them turned on to another Adrawing.

Each Adrawing provides two different view states in which you can view the model instances of the assembly. The default Standard Adrawing shows the model instances in their “true” or “assembled” position. The Alternate state (Exploded state) allows you to view model instances at some other location and/or orientation. Viewing model instances in the Exploded view state does not change their true location in the assembly. The Exploded view state allows you to test points of stopped motion. Showing a door in an opened position instead of its closed and locked position, or showing an exploded, isometric view of a subassembly for illustration purposes are two such examples.

Clarifying Terms

To better understand the model geometry in CAMU, you should be comfortable with the terms used. They are:

- Adrawing - a CADDS part, subordinate to the CAMU assembly database file, that enables you to see the model geometry associated with various components in your assembly structure (see “About Adrawings” on page 5-5).
- Model - a CADDS part that has been associated with a component class in the assembly structure.
- DDRAW - a CADDS drawing file, activated automatically when you activate an Adrawing. It is used only by a CAMU Adrawing and only for the purposes of containing and preserving a view state. A view state includes the following:
  - Number, location, scale, and orientation of views
  - Specific layer visibility
  - Specific model geometry display based on user-visibility choices for specific component instances, subassemblies, and so forth.
  - Specific visible layer ranges for user-specified component instances, subassemblies, and so forth.
  - A user-specified alternate view state, for example, an exploded view
  - Any annotation geometry that you have added to enhance the display of viewed model geometry in the view state, for example balloon callouts, text, and reference dimensions for an exploded subassembly (see “DDRAW” on page 5-7)
- MDRAW - a CADDS drawing file, activated automatically when you activate a model. It is used only by CAMU to enable seamless view display when moving from (being active in) an Adrawing to a model or from one model to another. An MDRAW view state is always inherited and never preserved (see “MDRAW” on page 5-7).

Moving Between Models and Adrawings

Moving between active and suspended Adrawings or models or both is called node switching. This enables “seamless” movement between Adrawings and models, such that what you see in the graphics window before the switch (the number of views, scale, orientation of views, viewed model instances, view render mode setting, spaceball and mouse-based dynamic view manipulation settings, and clip-model settings) is identical to what you see after you activate the model.

Please note: You activate a model using the ACTIVATE MODEL command or property sheet. You activate an Adrawing using the ACTIVATE ADRAWING command or property sheet.

When you move from an active Adrawing or model, the currently active Adrawing or model is suspended. You can then move to:

- A new Adrawing or a new model
- A suspended model
- A suspended Adrawing

What You See in the Graphics Window

What you see in the graphics window depends on which Adrawing was active the last time you worked in the assembly. If you activate a model, the view state of the active model inherits what was last viewed from an Adrawing or from a model that was previously active.

DDRAW

A DDRAW is a special CADDS drawing file that is automatically activated when you activate an Adrawing. The DDRAW maintains the view state of the latest filed version of a particular Adrawing.

MDRAW

An MDRAW is a special CADDS drawing file that is automatically activated when you activate a model from within CAMU. The MDRAW preserves the view state (orientation, scale, number of views, viewed model instances, view render mode setting, spaceball and mouse-based dynamic view manipulation settings, and clip-model settings) of the assembly when you move from one active model to another.

The MDRAW maintains the view state of the assembly only until you activate an Adrawing. When you activate an Adrawing, the view state (orientation, scale, number of views, viewed model instances, view render mode setting, spaceball and mouse-based dynamic view manipulation settings, and clip-model settings) resets to the view state of that particular Adrawing.

Please note: Detailing and other view-specific operations such as line fonting, dimensioning, and hidden line removal should not be attempted while working in an MDRAW.

The MDRAW has the ability to inherit the view state of the previous active drawing (MDRAW or DDRAW). In other words, the views contained in the MDRAW are temporary for the active session.

Any view-specific operation in a multi-view MDRAW can cause the corresponding entities in other views to be affected. The purpose of an MDRAW is to provide a seamless, visual transition from one active model to another.

ERASE ENTITY Design Restriction

All views in the MDRAW use the same (universal) composition. As CADDS entities declare their view-specific appearances via composition numbers, all appearance commands in the MDRAW effectively have the 'allviews' functionality. Manually activating a 'normal' drawing (anything other than MDRAW) in the CADDS model after node-switching allows you to use view-specific appearances.

Please note: You can use commands such as ERASE ENTITY, CHANGE APPEARANCE, and so on, if the entities are physically located in the active part.

Moving to a New Adrawing or New Model

- The drawing DDRAW (in the new Adrawing) or MDRAW (in the new model) is activated.
- No view/orientation/zoom/scale is inherited into the DRRAW of the new Adrawing. The existing view state of the previously active part (model or Adrawing) is inherited by the MDRAW of the new model.

Moving to a Suspended Model

The drawing MDRAW is activated in the suspended model. The view, zoom, and scale settings of the model is inherited by the MDRAW. The MDRAW views are oriented to preserve the assembly. This action is also called “seamless visual movement”.

Moving to a Suspended Adrawing

The view state of the suspended Adrawing is restored. There can be visual changes when activating a model based on the layer color choices of the active model, and so forth.

Understanding the Suspension Buffer

When another part (model or Adrawing) is activated, the previously activated part is suspended. The CADDSMAX_PART variable in the `.caddsrc` file specifies the maximum number of parts that can be suspended. The default is 6. The active part (model or Adrawing) counts as one (1). For example, if CADDSMAX_PART=9, you can suspend 8 parts (models or Adrawings) and have one active (see “.caddsrc-local File Information” on page 2-3).

To clear the suspended status on a given part, file (save) it. Filing empties the suspension buffer and allows you to activate or suspend, or both, additional parts. For more information see Chapter 4, “Filing an Assembly.”

Filing an Active Model That Is in History Replay Mode

You can discard or save the results of an edited or incomplete history replay session to the model database, for a parametric model associated with a component of an assembly.

If you are in the history replay mode of an active model, and try to activate another model or Adrawing, the following menu is displayed:

You can choose from the following options:

- File: Saves the results of an edited or incomplete history replay session to the model database by filing the model. The results are stored in the `_fd.edt` file of the part database.
- Quit: Quits the history replay mode, while retaining the part database files. The`_fd.edt` file is also retained if it exists.
- Remove: Deletes the existing `_fd.edt` file from the part database and quits the active model.

Examples of Moving Between Adrawings and Models

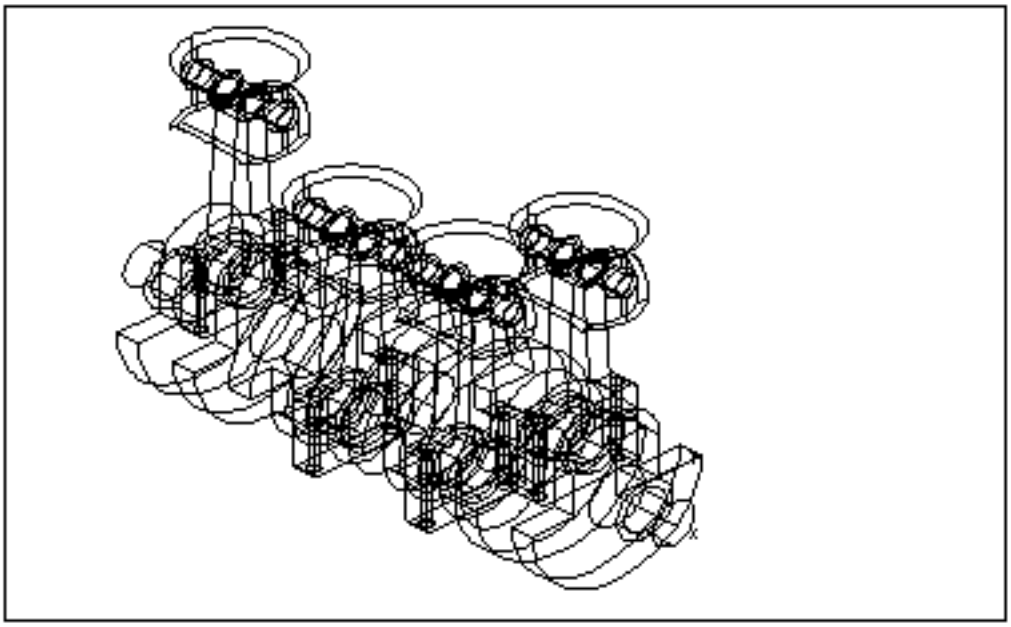
The examples that follow demonstrate what happens in both the Assembly Structure window and in the CADDS Graphics window when you move between Adrawings and models.

- Activating the Assembly and an Existing Adrawing
- Activating a Model While Active in an Adrawing
- Moving from One Active Model to Another
- Moving from an Active Model to an Active Adrawing
- Activating a New Adrawing
- Moving from a New Adrawing to an Existing Adrawing

Activating the Assembly and an Existing Adrawing

When you activate an existing assembly named `new-crankassy` and activate an existing Adrawing in which the view state is already defined, the Adrawing has the model instances of the entire crankshaft assembly viewed in the isometric orientation. Although the model graphics of the crankshaft are visible in the graphics window, you cannot change any of the model geometry because you have activated an Adrawing, which is also a CADDS part.

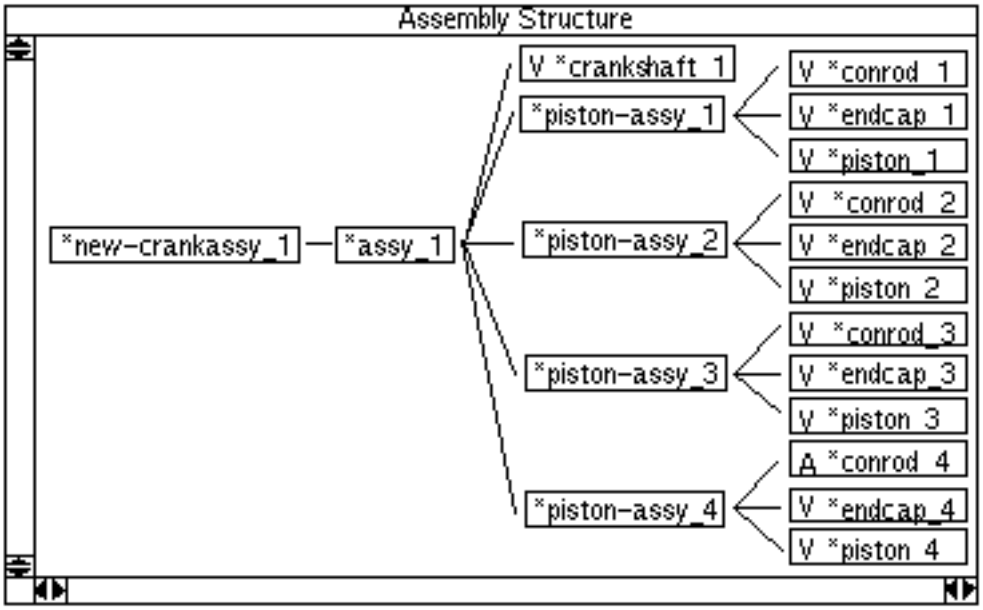
Figure 5-2 Graphics Window with the Crankshaft Assembly in ISO Orientation



Activating a Model While Active in an Adrawing

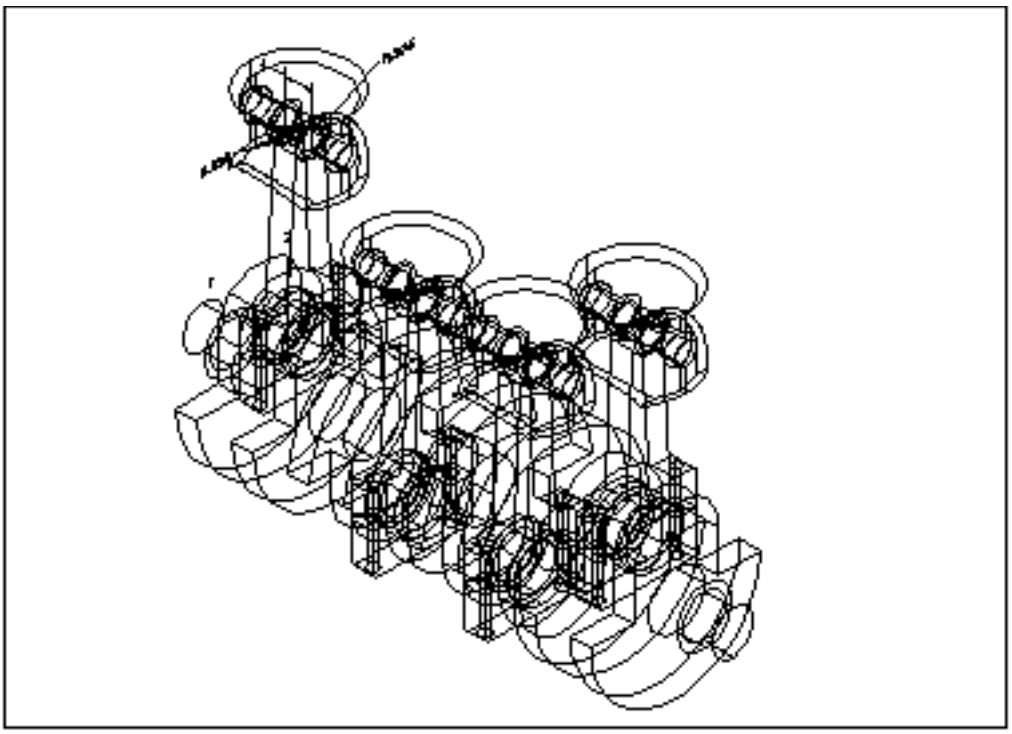
When you activate the model associated with the component instance `conrod_4`, the letter “A” appears in the `conrod_4` component box, indicating that the associated model is active.

Figure 5-3 Assembly Structure after Activating the Conrod_4 Model Instance



The view state of the assembly in the graphics window does not change.

Figure 5-4 Graphics Window after Activating the Conrod_4 Model Instance



Moving from One Active Model to Another

When you activate the model associated with the component instance `piston_4`, activating another model suspends the model associated with `conrod_4`, which was previously active. An icon appears next to all instances of `conrod`, indicating that their associated model is suspended. Activating another model does not change the view settings such as the view orientation, scale, number of views, viewed model instances, view render mode setting, spaceball and mouse-based dynamic view manipulation settings, or clip-model settings in the graphics window.

Please note: Activate a suspended model by selecting the component instance to which it is associated. Suspending previously activated models lets you experiment with changes in the context of the assembly, and decide later which changes you want to save or discard when you file the assembly.

Figure 5-5 Graphics Window after Activating the Model Instance Piston_4

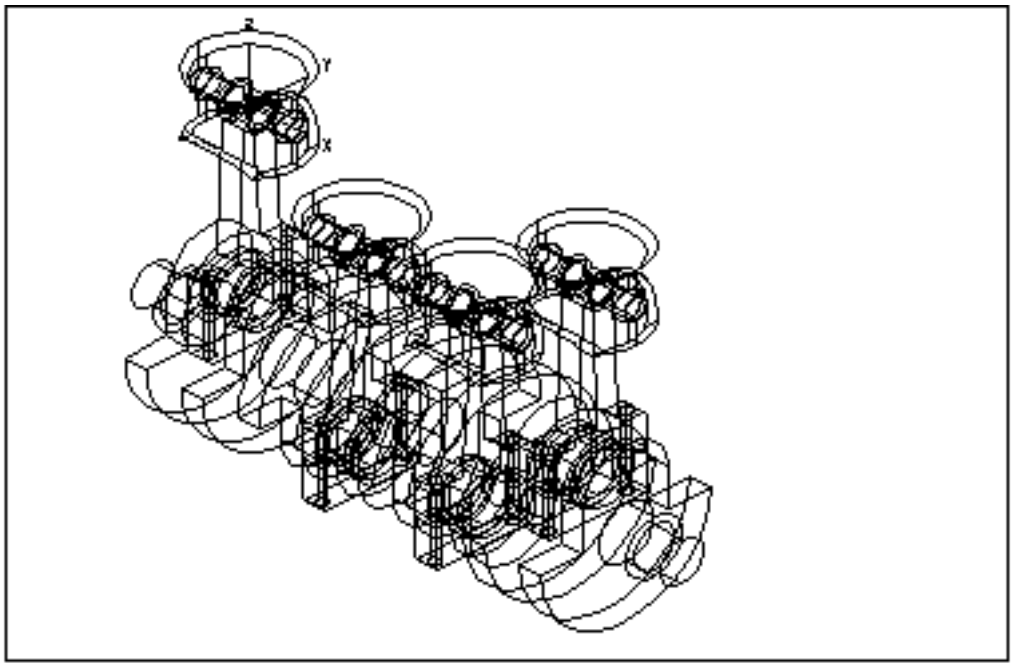
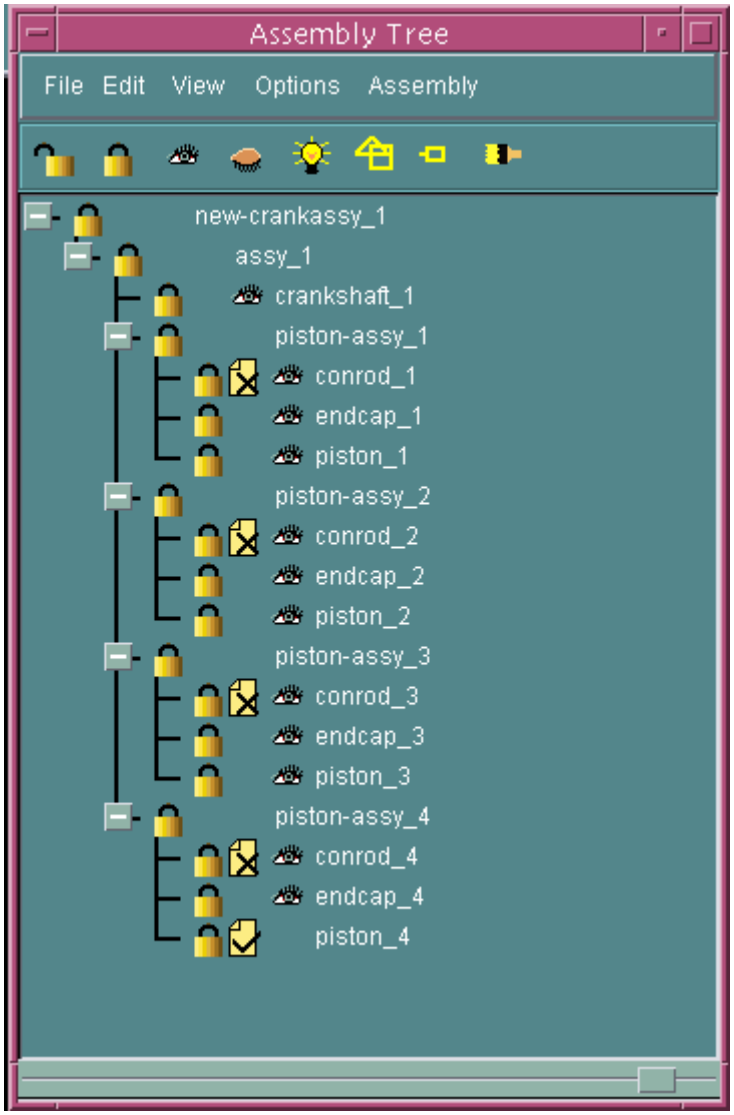
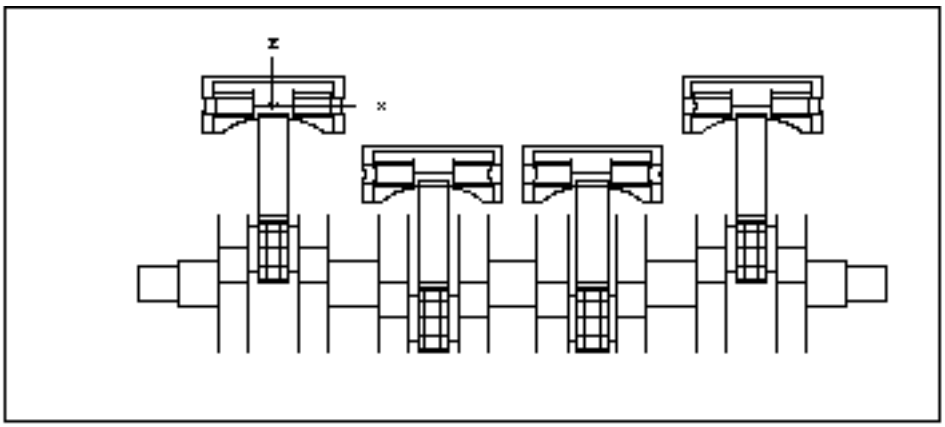


Figure 5-6 Assembly Structure after Activating the Model Instance Piston_4



Changing the View Orientation to Front: The view orientation is changed to Front to further demonstrate how the MDRAW preserves the view state of the assembly in the graphics window, when moving from one active model to another.

Figure 5-7 Crankshaft Assembly Changed to Front Orientation with the Piston Model Active



Activating a Model After Changing the View Orientation: When you activate the model associated with crankshaft with the assembly viewed in the Front orientation, the previously activated models, conrod and piston, are both suspended.

Figure 5-8 Graphics Window after Activating the Crankshaft Model Instance

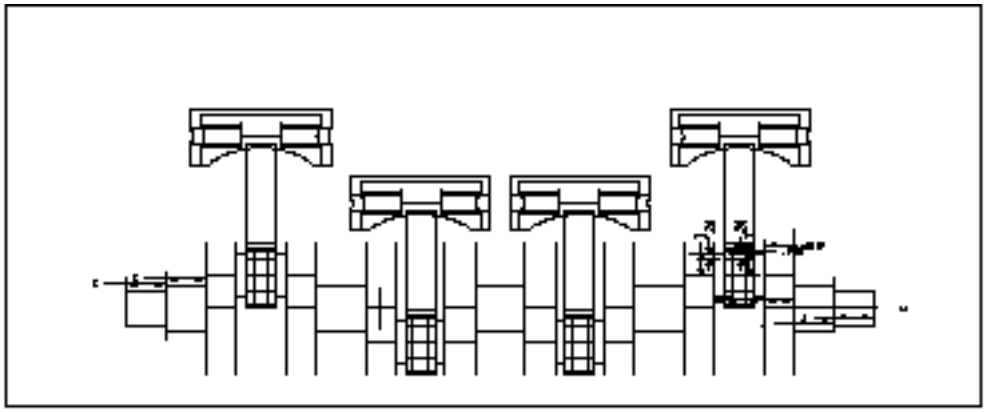
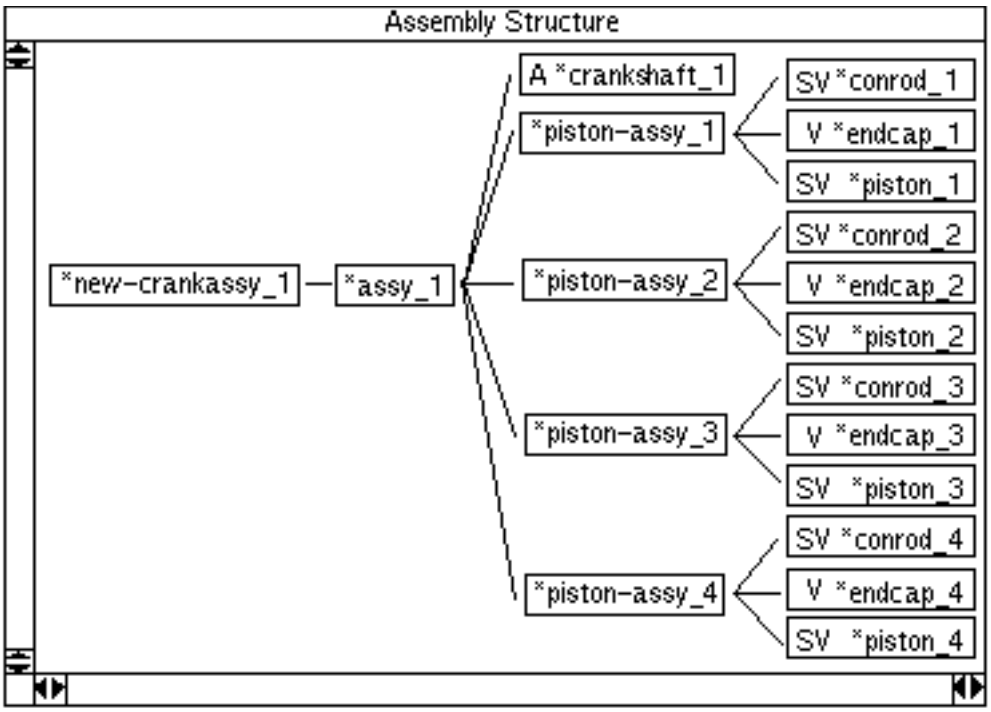


Figure 5-9 Assembly Structure after Activating the Crankshaft Model Instance



After activating a new model, your view of the assembly in the graphics window does not change. The number of views, orientation, zoom, scale, viewed model instances, view setting, spaceball and mouse-based dynamic view manipulation

settings, and clip-model settings are identical to the view state of the assembly before you activated another model.

Please note: When moving from one active model to another active model, the view state of the assembly is preserved by the MDRAW. This enables you to activate and work on the geometry of different models, in the context of the assembly, without disrupting your view of the assembly itself.

Moving from an Active Model to an Active Adrawing

When you activate an existing Adrawing, all the previously activated models are suspended. After activating an Adrawing, the view state in the graphics window changes. This is because Adrawings have their own view states which are preserved by DDRAW. DDRAW preserves the view state of the latest, filed version of the Adrawing.

Figure 5-10 Graphics Window after Activating an Existing Adrawing

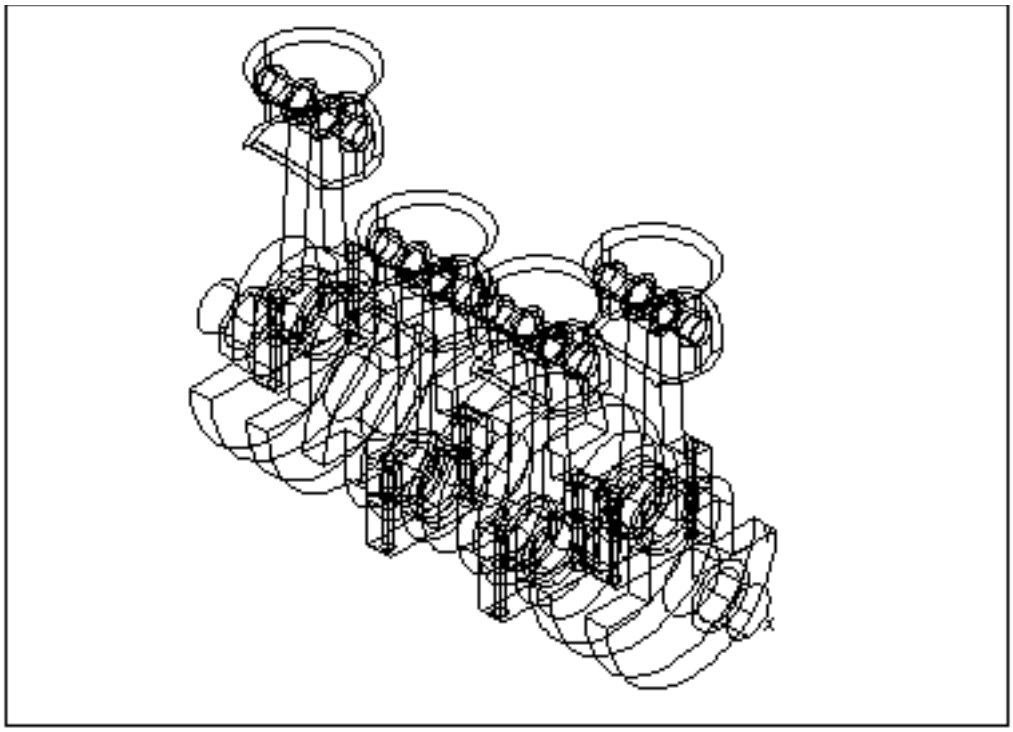
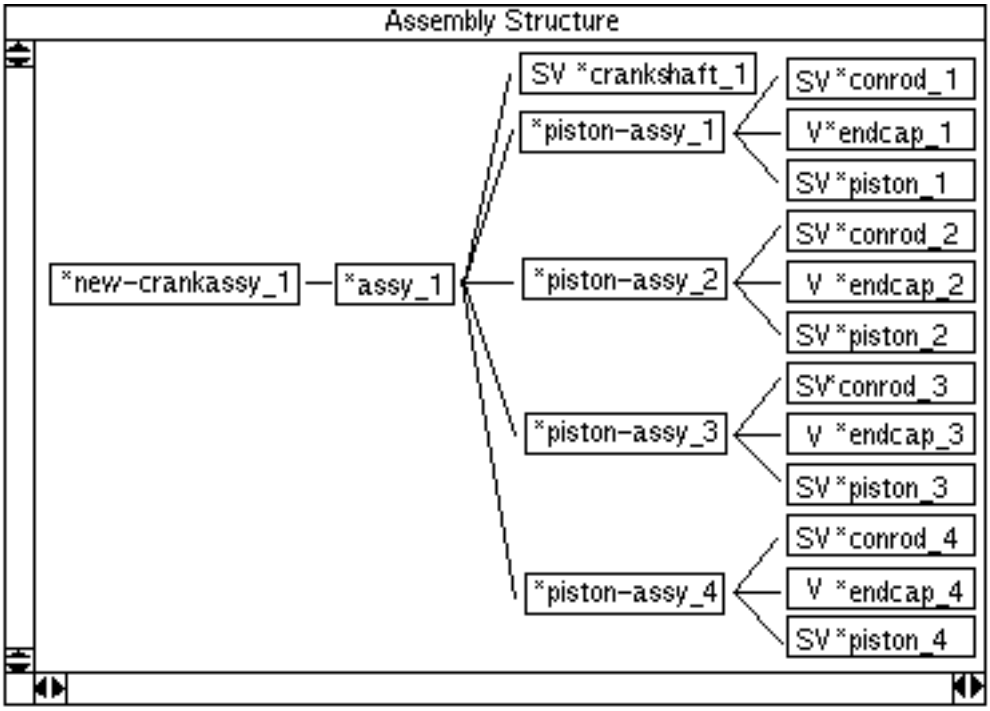


Figure 5-11 Assembly Structure after Activating an Existing Adrawing



Activating a New Adrawing

When you activate a new Adrawing, the model instances previously viewed in the graphics window are no longer visible.

Figure 5-12 Graphics Window after Activating a New Adrawing with Four Views

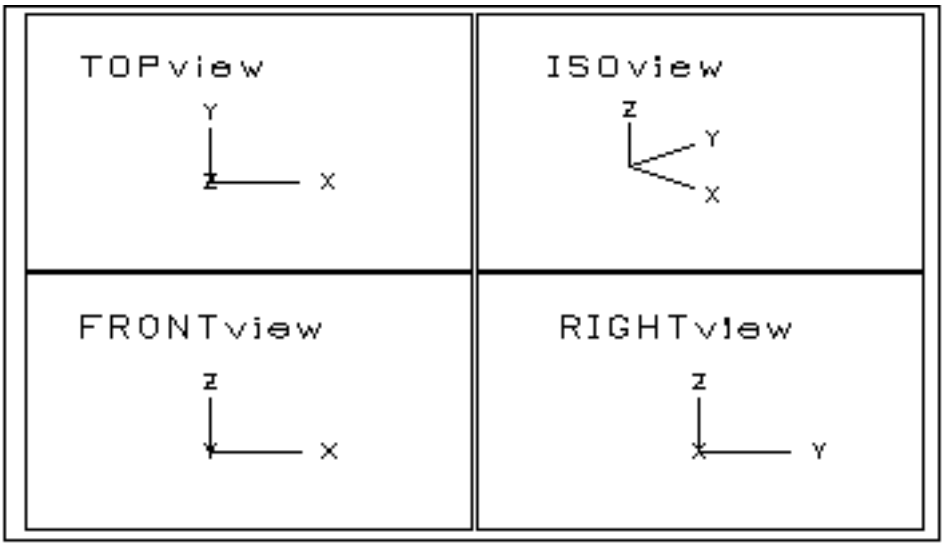
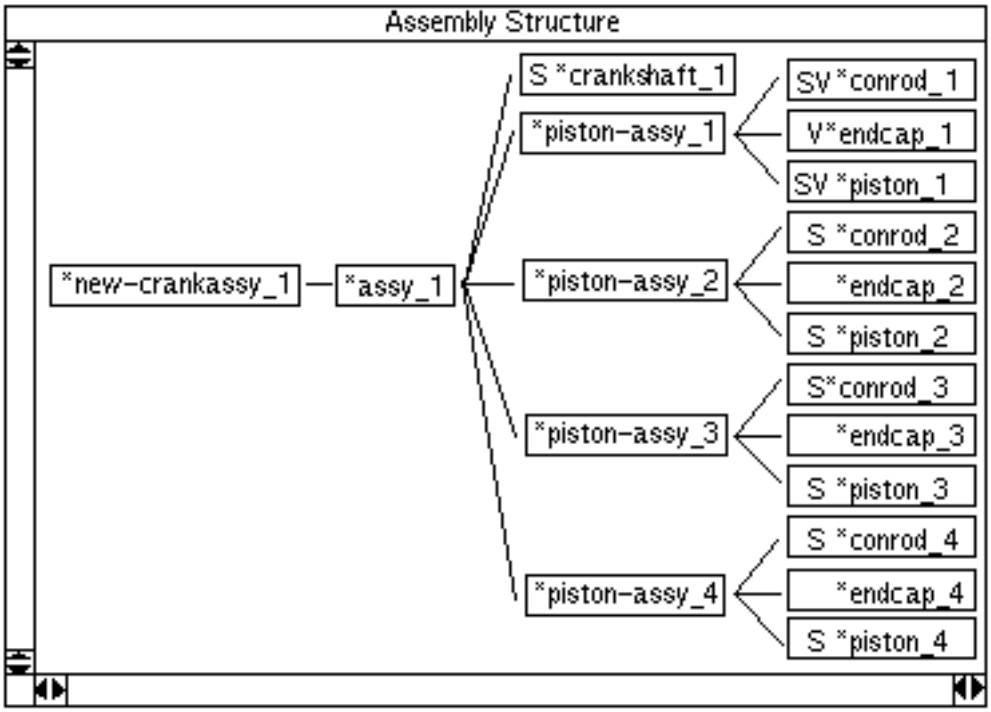
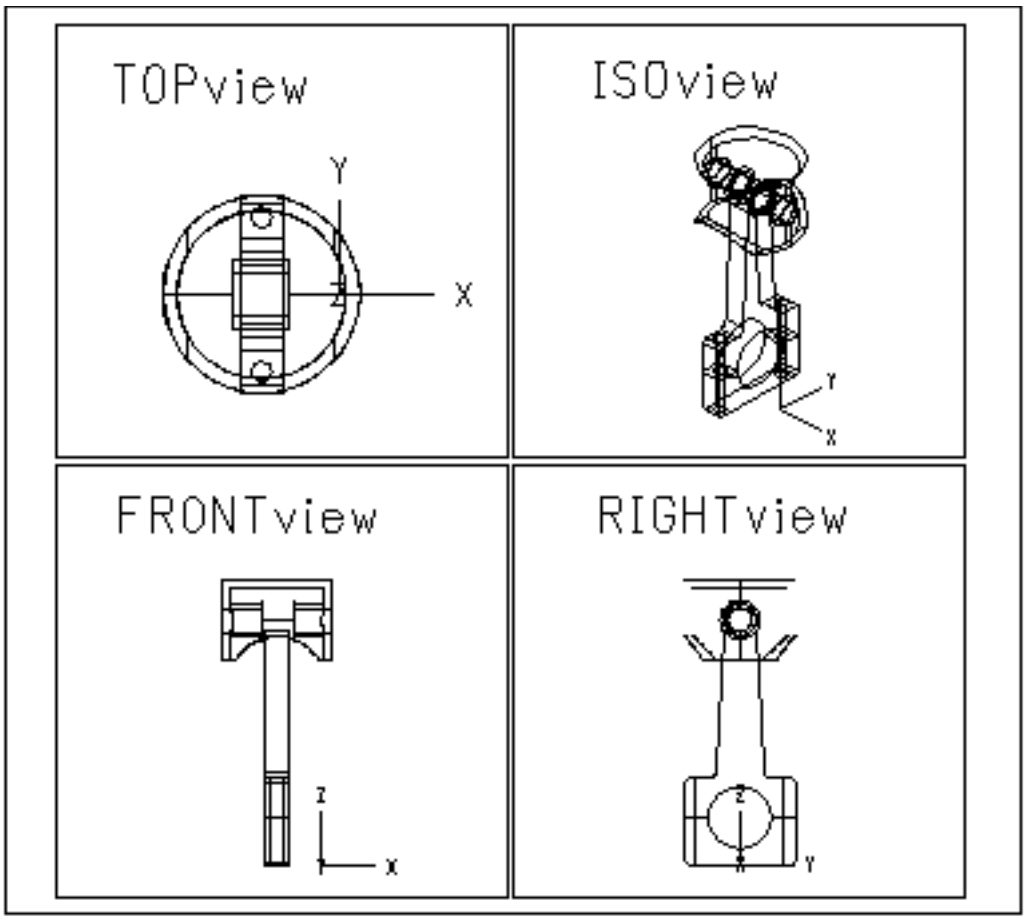


Figure 5-13 Assembly Structure after Activating a New Adrawing



Viewing Model Graphics in the New Adrawing: You can view the component instance `piston-assy_1` to view the model graphics associated with its leaf components. To view the parametric piece parts, turn on the display of all layers. The `piston-assy` leaf component model instances will now be visible within all four views in the graphics window. In the Assembly Structure window, all previously activated models are suspended.

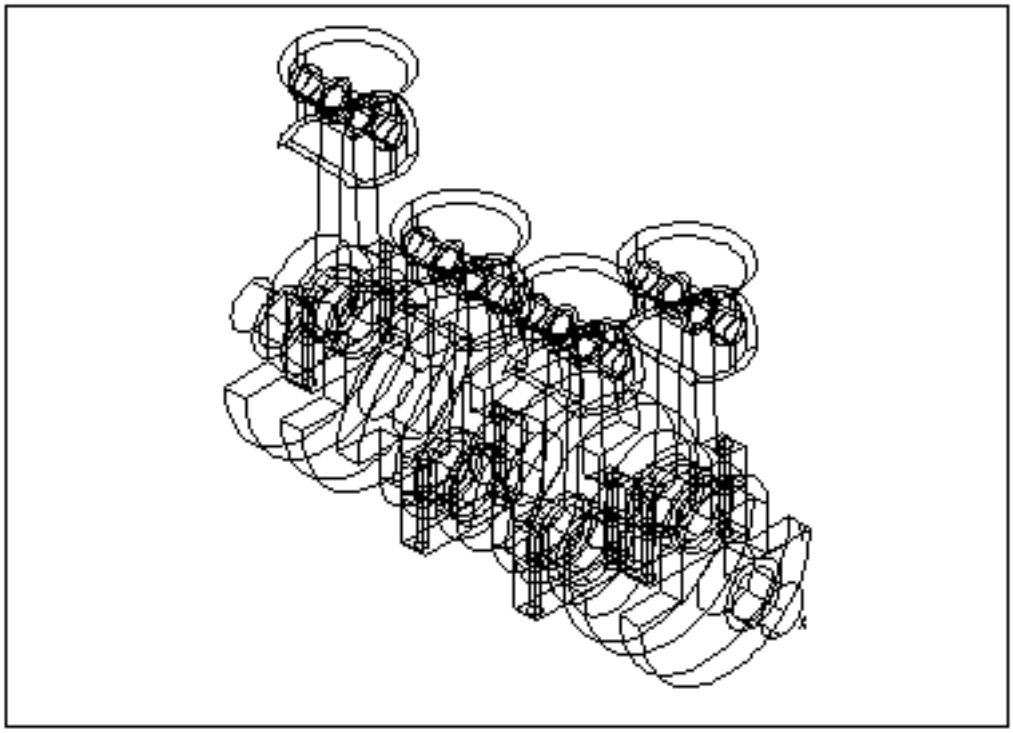
Figure 5-14 Graphics Window with piston-assy_1 Viewed in the New Adrawing



Moving from a New Adrawing to an Existing Adrawing

When you activate the existing Adrawing named "1", the graphics window shows the view state of the assembly as it was last viewed (and saved) on this Adrawing.

Figure 5-15 Graphics Window after Moving from a New Adrawing to an Existing Adrawing



Visual Changes When Activating Parametric Models

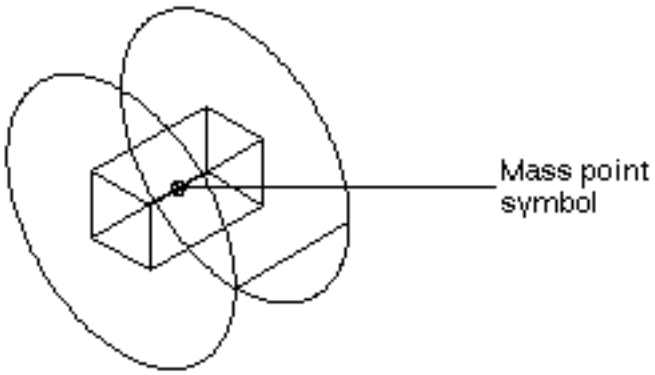
When activating Parametric models or piece parts or both, you may see the following changes in the graphics window:

- The TOP Cplane of the active part may be different for each model or piece part. Therefore, its position may change in the graphics window when you activate a different model.
- The color of the active model or piece part geometry may change the graphics window if the layer assignment at the piece part level is different from the layer assignment on which it is viewed in the active Adrawing.
- Any parameters that were visible the last time the piece part was filed appear in the graphics window when you activate the model.

Associative Mass Points

This section explains how to work with mass points, and how to report mass point data in the Explicit environment.

Mass points provide a way to store physical properties data in the Explicit environment. Each time you perform a calculation, you can create a mass point. The system uses the standard center of mass symbol to identify the location of the object's centroid (center of mass). You can change this location with the Move icon.



You can also store physical properties data outside the part database in a standard text file. This allows you to quickly transfer physical property data to and from mass points.

For more details about storing physical properties data, see the *Physical Properties User Guide and Menu Reference*.

Mass points can be associative or nonassociative. Mass points can optionally be merged to create composite mass points to determine weight, center of gravity, and so on of multiple objects. The mass point is inserted into the physically active part.

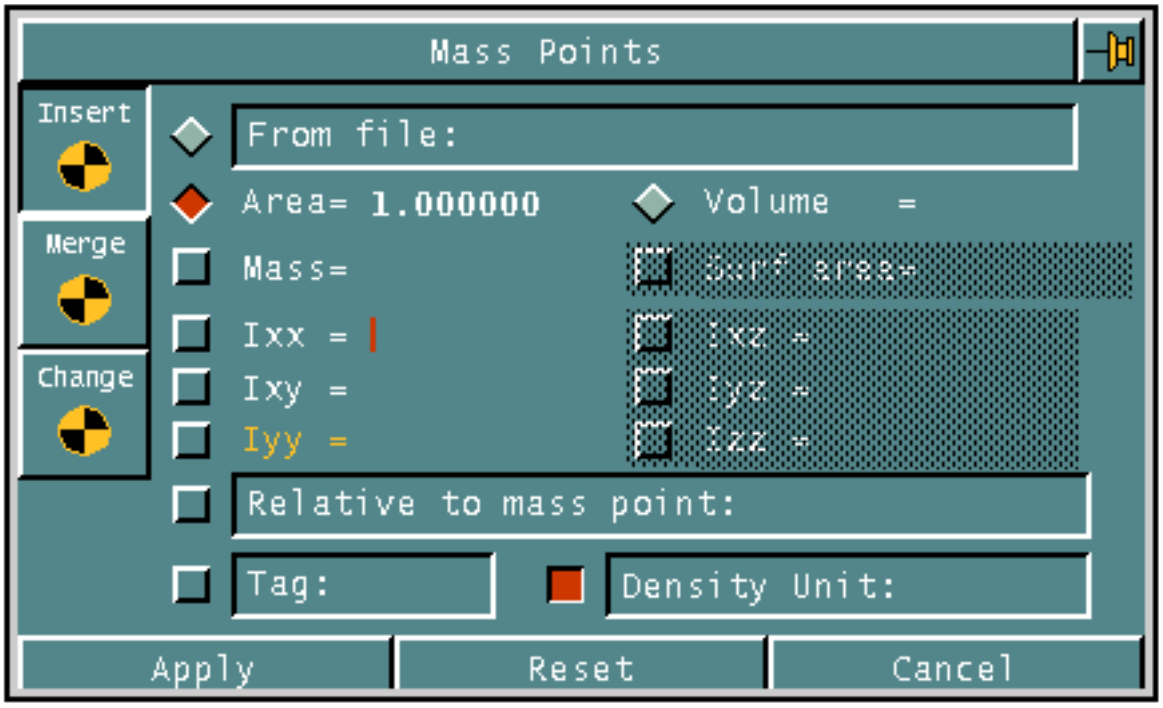
Please note: Associative mass points can be created only if the solid or Tsurf to which they pertain is physically resident in the active part. As opposed to this, the ASSOCIATIVE option in the CAMU environment is available only with the ADD option. There is no such limitation in the active part environment.

Using the Associative Mass Points Option

Select the Physical Properties icon from the Model task set.

The following command palette is displayed.

Select the Mass Points icon from this palette to display the following Mass Points property sheet.



This property sheet allows you to insert, change and merge mass points. Selecting any one of these icons changes the property sheet to the relevant modifiers.

Select the Update Select Associations icon to update one or more associative mass points. These mass points are based on the most current size, shape, and so forth of either the associated solid, the trimmed surface, or data of the other associative mass points (source mass points).

Select the Update All Associations icon to update all associative mass points in the active part.

Select the Mark Mass Points icon to highlight mass point associations by marking the entities that are associated with the selected object.

Select the Layer icon to move the highlighted entities onto the specified layer.

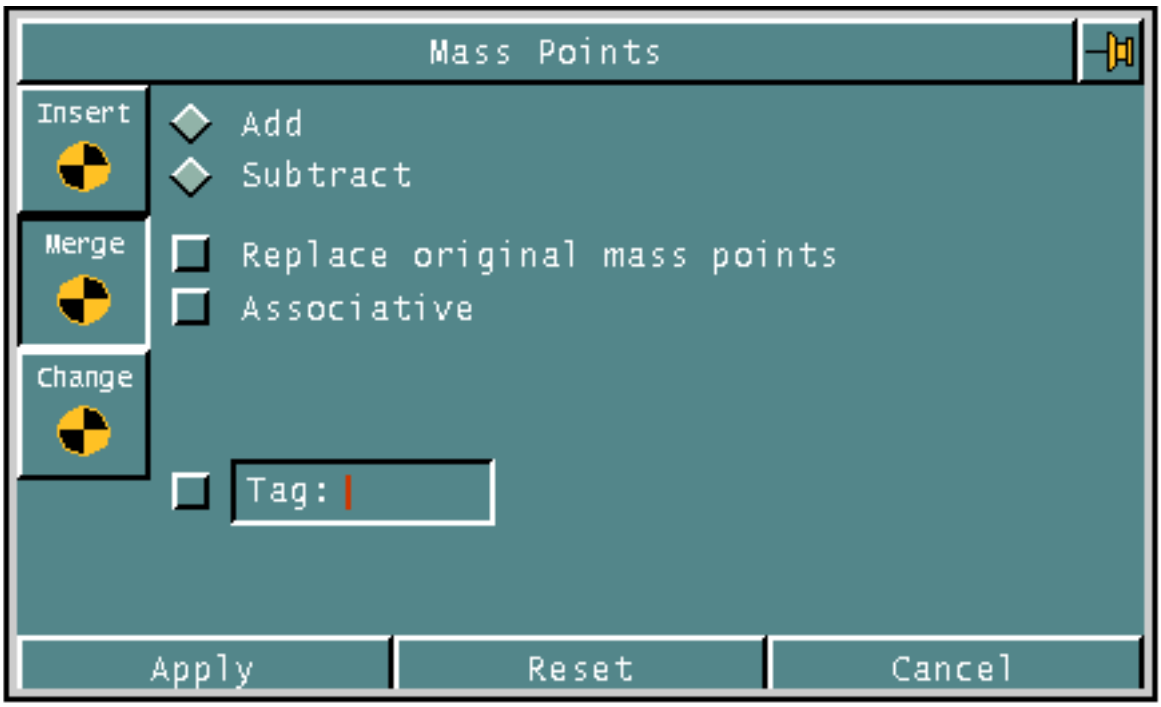
Select the Dissociate icon to break the association between a solid or Tsurf and its associated geometric mass point or between a merged mass point and all of its source mass points.

Using Merge Mass Point Options

In the CAMU environment, the ASSOCIATIVE modifier is available only with the ADD modifier. When some of the selected Source Mass Points are viewed-in, a new Merged Mass Point is created without replacing the existing one. To differentiate between the new and the old mass point, use the MARK MPOINT, UPDATE MPOINT and DISSOCIATE MPOINT commands to determine the relationships.

Select the Merge option on the Mass Points property sheet. The property sheet changes as shown. The Merge option allows you to calculate the mass properties of an object, for instance the sum or difference of two or more other objects. For example, to calculate total weight, merge the mass points of all models within an assembly.

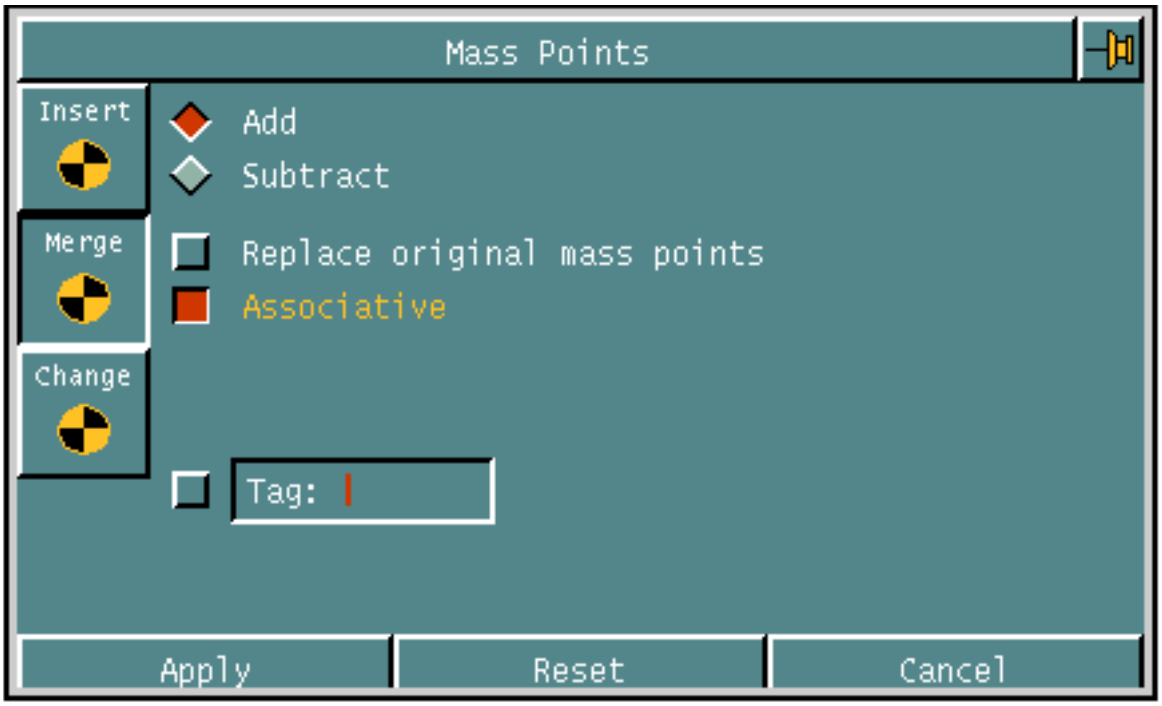
Figure 5-16 Merge Mass Points property sheet



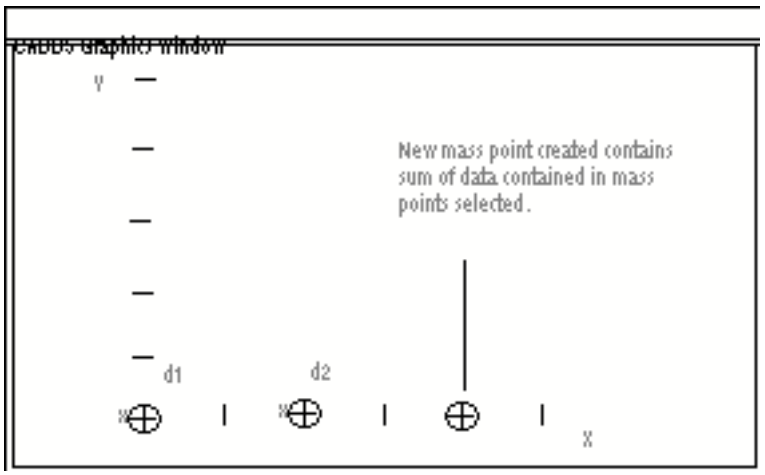
- Choose Add to add the indicated mass points.
- Choose Subtract to subtract the second and subsequent mass points from the first one.
- Choose Replace original mass points to delete the original (source) mass points of the merged mass point.
- Choose Associative to create an association between the new merged mass point and all selected source mass points.
- Choose Tag and assign a name to the mass point. Tag names are limited to six alphanumeric (A-Z, 0-9) characters.

The new merged Mass Point is created in the active part, which is typically an Adrawing. This mass point is associated with the selected source mass points. The source mass points are in the viewed-in models.

The following example creates a mass point whose values represent the sum of two merged mass points.



1. Select the Merge option on the Mass Points property sheet.
2. Choose the Add option.
3. Now choose the Associative option.
4. Click Apply.
5. In the graphics window, select the mass points that you want to merge.



6. Select Done from the Utilities menu.

Several Users Can Work Concurrently

A critical and central feature of CAMU is its ability to enable multiple users to work simultaneously on the same assembly database (`_db`). Several users can work on the same assembly at the same time. You must lock the component(s) on which you want to work. Locking is central to the concurrent assembly design process (see "Locking Components, Attribute Groups, and Models" on page 5-31).

Please note: To configure the system for multiuser operation, see Chapter 2, "Configuring Concurrent Assembly Mock-Up".

Determining Status in Multiuser CAMU

VERIFY OBJECT displays whether the component is active or suspended, and if so, by which user.

LIST ADRAWINGS display the details of Adrawings in the assembly, including which user has write permissions to a particular Adrawing.

The CAMU Status window states the name of the selected component, its associated model name, the name of the user who has locked the component, and its status (inactive, active, or suspended).

Updating Your Session

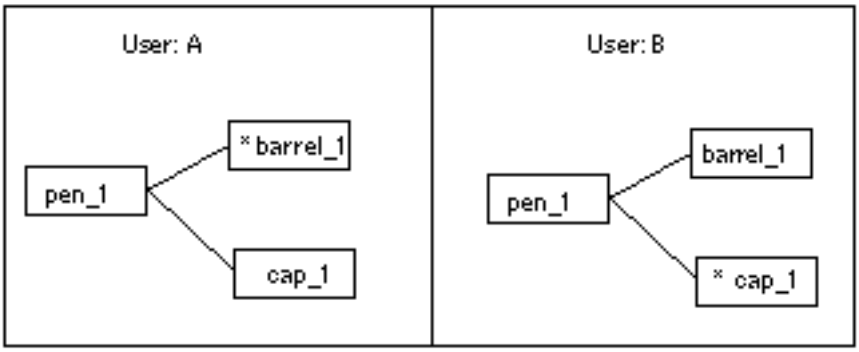
File or refresh the active main assembly to see changes that other users have made and filed in their CAMU session. This also updates associative dimensions in the assembly. See Chapter 4, "Filing an Assembly" and "Refreshing the Main Assembly" on page 5-36 for more information.

See "Updating Positions After Modeling Change" on page 6-37 for more information on updating positions parametrically after modeling changes have been made.

Multiuser CAMU Example

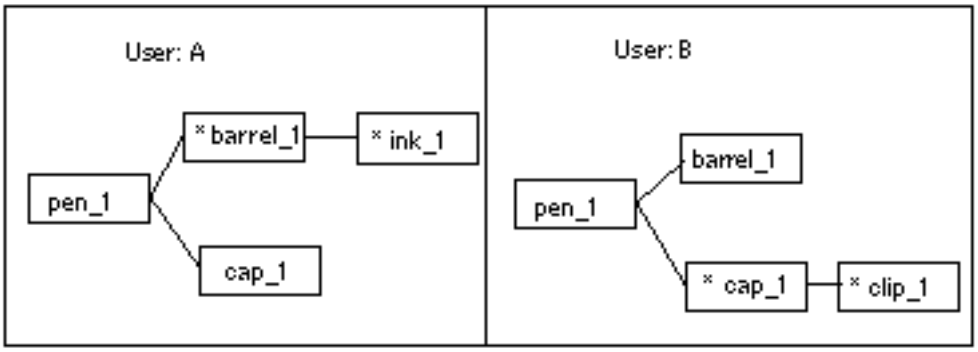
In the example, two users, A and B are working concurrently on the pen_1 assembly. A is working on the `barrel_1` assembly and B on the `cap_1` assembly.

- A and B activate `pen_1` assembly.



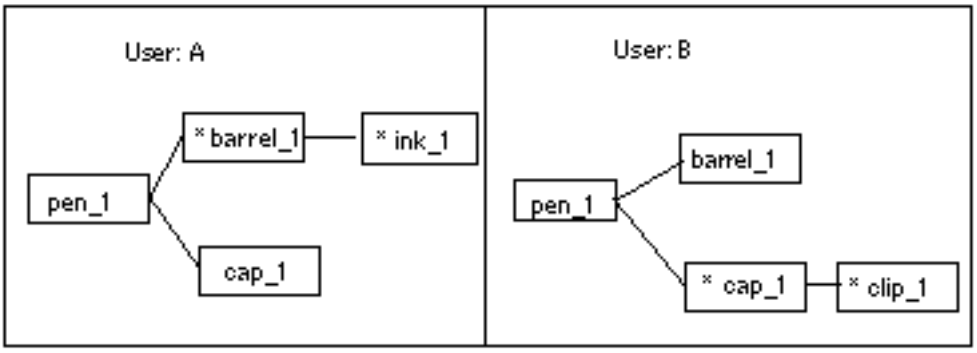
A and B both activate the `pen_1` assembly, and each of them lock the portion on which they are working. A asterisk (*) indicating the component is locked appears in the applicable component box for each user.

- A and B begin modifying the assembly.



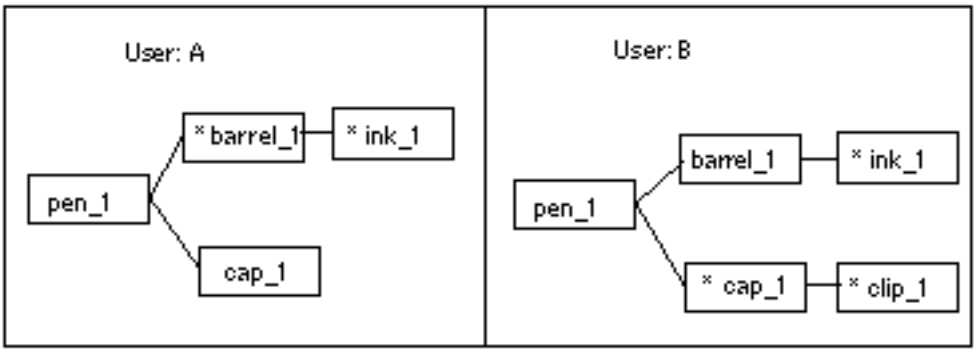
A adds an ink cartridge to the `barrel_1` component. B adds a clip to the `cap_1` component. Neither B nor A have filed or refreshed the assembly, so neither see what the other has done.

- A files the assembly.



A selects the FILE icon and selects File Assembly. This commits him changes to the database. Note that what A sees is the same as that before filling, because B has not yet filed the work.

- B refreshes the assembly.

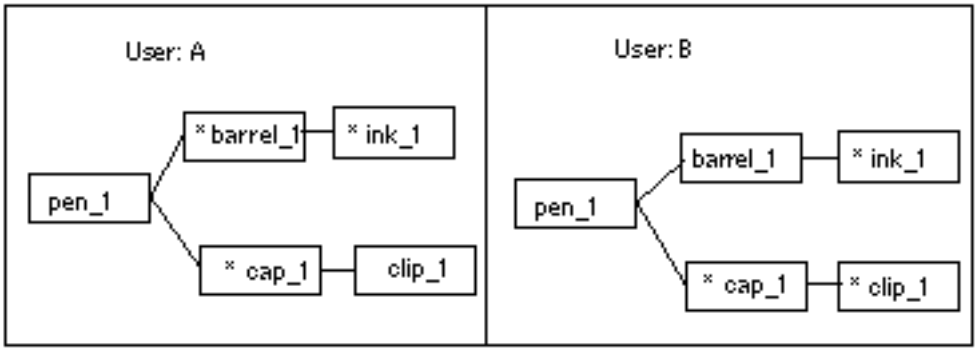


B selects Refresh Assembly. It allows him to see the changes that A filed. However, the refresh has not committed any of B's changes to the database. This means that if A files or refreshes, he cannot see B's changes.

If, instead of refreshing the assembly, B had filed the changes, they would be saved in the database. A refresh would have been done automatically for him, showing A's changes. The figure above would look the same, but B's changes would be saved.

A cannot see B's changes until B files, and A files or refreshes the assembly.

- B files; then A refreshes the assembly.



B selects File Assembly, committing changes to the database.

When A refreshes, then only sees B's changes.

Updating Parametric Positions After Modeling Changes

If the assembly contains parametric positioning constraints, see "Updating Positions After Modeling Change" on page 6-37, the users must file the model and the assembly before their effects can be seen by other users.

For example, A and B are working concurrently in an assembly. A parametrically positions several parts. A activates one of the parts, changes some of its parameters, regenerates the model, and files the model. A then files the assembly in order to see the positional effects of the modeling change. B sees the changes when he either files or refreshes the assembly.

To facilitate this scenario, set the following in your `.caddsrc` or `.caddsrc-local` before invoking a CAMU session:

```
setenv CADDSASSOC_UID "yes"
```

Checking Which Server Process is Attached to the Assembly

With several users working concurrently on the same assembly, there may be times when you want to determine which ODB_SERVER process is attached to a particular assembly. To do this, open a shell window and enter the command:

```
/bin/ps -ef | grep ODB_SERVER | grep -v grep
```

See Chapter 2, "Configuring Concurrent Assembly Mock-Up" for more information on CAMU system management.

Locking Components, Attribute Groups, and Models

Locking establishes read/write permissions on the assembly (or portions of it), attribute groups, and component model geometry. It is a safety mechanism that prevents users from activating and filing changes to the model associated with the assembly component at the same time. Its purpose is to prevent users who are active in the same assembly from having write permissions to the same components at the same time. This prevents work from being inadvertently overwritten.

Components are automatically locked when you create them. However, when you activate an existing assembly, nothing is locked. You must lock the portion of the assembly on which you want to work.

Locking is session-dependent. When you exit the assembly, all locks are removed. Only the person who has locked an item can unlock it. You cannot lock an item that another user has locked.

Lock Component data is passed to an auxiliary process called ODB_SERVER, which is attached to your assembly upon activation. It keeps track of who has locked what in the active assembly.

In the single-user environment, simply lock the entire assembly, from the root node down. In the multiuser environment, you must be selective about what components you lock. Only lock what you need to work on. Locking a component gives you edit rights to it and prevents anyone else from locking it.

Please note: You can activate and edit a reference assembly from the active main assembly by locking a child component of the associated assembly instance. Locking any child of an assembly instance activates the associated assembly database (`_db`) (see “Editing an External Assembly” on page 7-27).

The asterisk symbol (*) indicates that the component or assembly instance is now locked by you.

Locking Areas of the Structure

When you lock a component or subassembly, other users have read-only access to it. They cannot activate and file changes to the model associated with the components you have locked. They cannot manipulate your locked component in any way.

Please note: When you lock a child component of a reference assembly instance you automatically activate the assembly database associated with the assembly instance.

An asterisk symbol (*) in the component box indicates that you have locked the component. No indicator appears in your Assembly Structure window if another user has locked a component. To obtain information on a component, use Verify Object to see who has locked the component.

If you lock one instance of a component, all other instances of that component class are also locked. This is because all instances of a class are associated with the same model. For example, locking the `wheel_1` component instance also locks `wheel_2`, `wheel_3`, and `wheel_4`. All instances of a class are clones of that class.

If you lock a portion of the assembly, then add, change, or remove components, you cannot remove the locks until you file the assembly. Anything new that you add to the structure is automatically locked. You cannot remove the lock until you save or quit.

Locking is not required for viewing a component's associated model. It is also not required for positioning (MOVE COMPONENT) items in an exploded Adrawing (see "Using Alternate Views in Your Design" on page 10-6).

Determining What to Lock

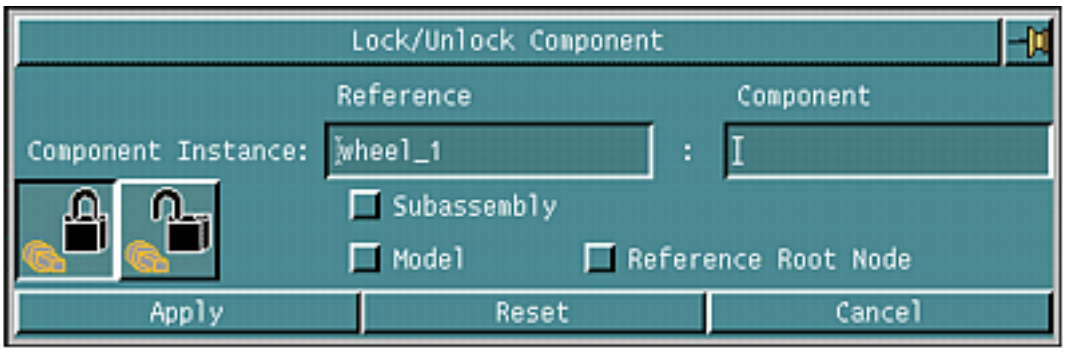
CAMU can automatically lock components for you based on the task you wish to perform. For example, if you want to add a new component to a selected parent, that parent must be locked. You can either lock the parent yourself (using Lock Component) (see "Lock/Unlock Component Property Sheet" on page 5-33) or let the system lock it for you as a function of the Add Component task (see "Adding a New Component to the Structure" on page 3-9).

Please note: Locking any component within a reference assembly automatically activates that reference assembly in the main assembly (see "Activating a Reference Assembly" on page 5-34).

Lock/Unlock Component Property Sheet

The Lock/Unlock Component property sheet has several options:

Figure 5-17 Property Sheet for Locking or Unlocking a Component



Reference: Specifies the name of the reference assembly (assembly instance) to of which the component is to be locked. Once specified, you can lock a Component within that referenced assembly. Leave this entry blank if you want to lock a component in the active main assembly.

Component: Specifies the component instance to lock within either the active main assembly or the named Reference assembly.

Subassembly : Locks the component instance and its children components. If other users are working on the same assembly, selecting this option prevents them from activating and filing any changes to the model geometry associated with this component.

The Lock Component command can lock components in a reference assembly to establish the right to modify those components. The Subassembly option will lock the selected component and its children until the first reference instance (see “Changing a Reference Assembly” on page 7-25).

Model: Locks the model associated with the named Component. If you have also specified the Subassembly option, it locks all models associated with the subassembly.

Please note: The Model option is not required to lock the component and activate the model associated with it. It prevents other users from activating the associated model and filing any changes made to it, including users who are not working in your assembly. Using this option places a `LOCK` and `TEMP` file in the associated model’s subdirectory.

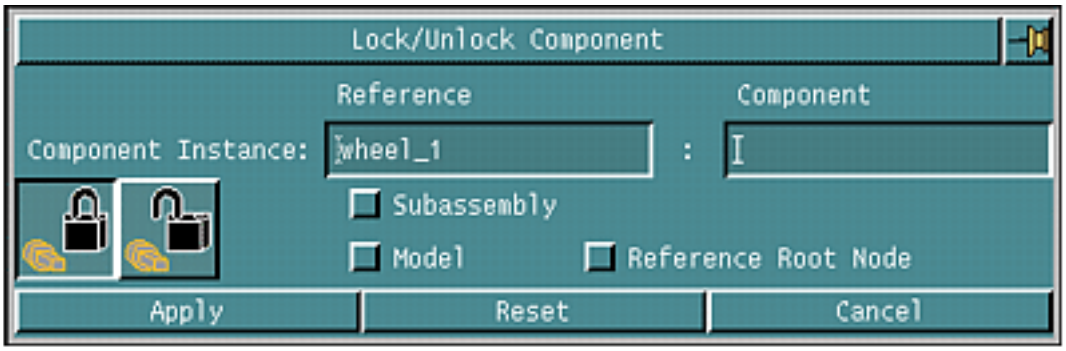
Reference Root Node Also : Locks the root node of the reference assembly.

Activating a Reference Assembly

The first time you lock any component within a reference assembly, the reference assembly is automatically activated (see “Editing an External Assembly” on page 7-27).

You can activate and save the changes made to the reference assembly without quitting the main assembly. The root node of the reference assembly and the reference instance are represented by the same node in the main assembly.

Figure 5-18 Lock/Unlock Component menu for an reference assembly

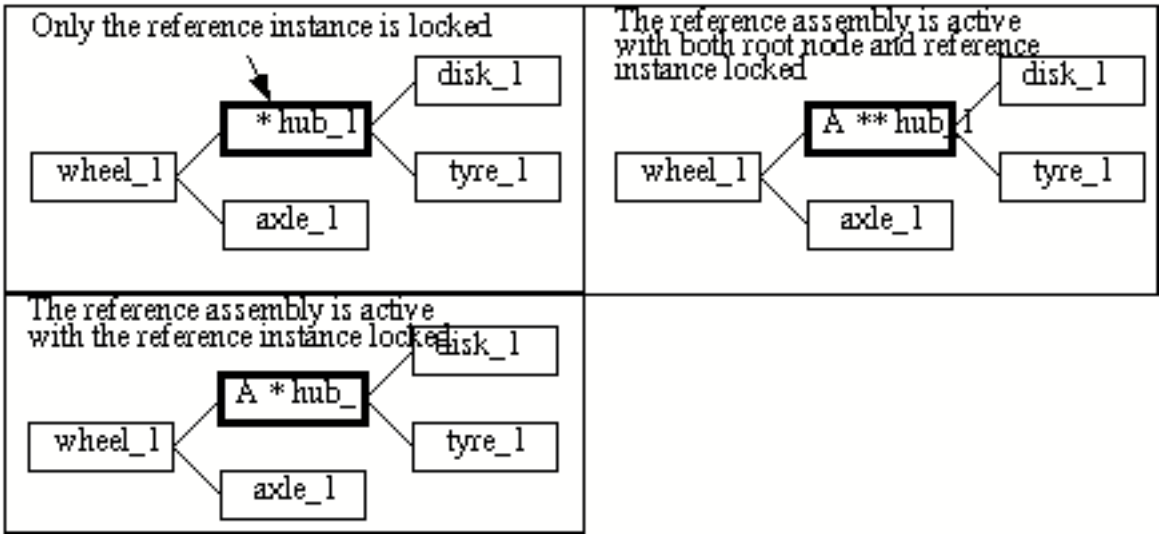


You can differentiate the lock status of the reference instance and the root node of the reference assembly in the assembly structure as:

- Lock on only the reference instance is represented with a ‘*‘ (asterisk) on the reference instance node.
- Lock on the root node and the reference instance of the reference assembly is represented with ‘**‘ (two asterisks) on the reference instance node.
- If the reference assembly is active, the character ‘A’ appears on the reference instance node.

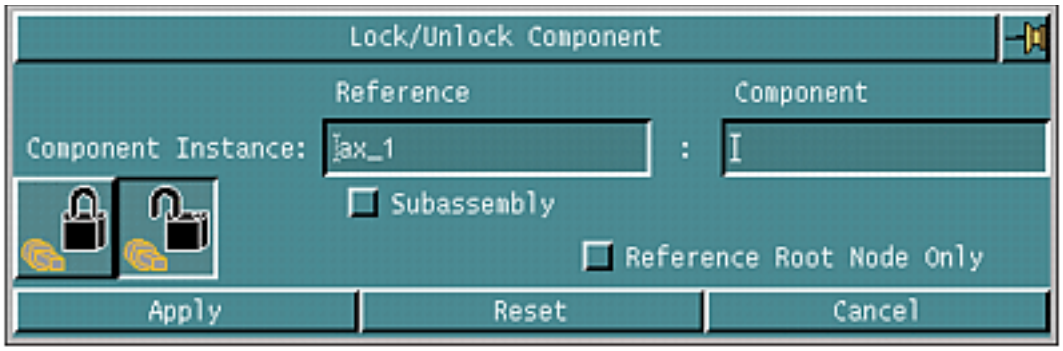
Please note: You can lock the root node of the reference assembly, only if you have the lock on the reference instance.

Figure 5-19 Different lock status of root node of a reference assembly



You can unlock the root node only or both the root node and the reference instance.

Figure 5-20 Unlock Component Property Sheet



Refreshing the Main Assembly

Refresh an active assembly to update its assembly structure, graphics, and associative dimensions.

When you refresh an assembly, changes made and filed by other users working on the same assembly appear in your active assembly. You can see the most current version of the assembly structure and any viewed geometry.

REFRESH ASSEMBLY can detect for reference assemblies that have changed in the current session but have been deleted by another user working concurrently in the same main assembly. This event opens an alert window with a prompt for filing or quitting the main assembly.

If the model associated with the root component is active when a refresh is performed, the constraints management system is updated to reflect changes from the imported components in the model associated with the root component. Models are not updated according to the new constraint parameters. For more information on constraints, see Chapter 9, “Using Interpart Constraints”.

Refreshing the assembly updates all associative annotation.

When you issue REFRESH ASSEMBLY, the following occurs based on the level of parametric constraints in place in the assembly. The Modify Position Constraints menu may appear (see “Modify Position Constraints:” on page 6-17).

Please note: If you have set Select Assemble On previously, all parametric positioning constraints are solved when you issue this command.

Fully Constrained Assembly

The current version of the assembly is displayed. This includes the newly computed positions of all components. See “Modify Position Constraints:” on page 6-17 for options and more information.

Underconstrained Assembly

The current version of the assembly is displayed. This includes an optimal solution for the positional constraints. Feedback is provided.

See "Modify Position Constraints:" on page 6-17 for more information.

Overconstrained Assembly

The current version of the assembly is displayed. This includes the newly computed positions of all components by eliminating the redundant constraints. If they are inconsistent, the system provides feedback.

See "Modify Position Constraints:" on page 6-17 for options and more information.

Chapter 6

Parametric Positioning

This chapter describes how to parametrically position components in the Parametric Assembly environment. The topics are:

- Understanding Parametric Positioning
- Creating Positioning Constraints
- Using Constrain Component
- Solving Parametric Positioning Constraints
- Changing Positioning Constraints
- Updating Positions After Modeling Change
- Using Pre-Revision 8.0 CADDS Models
- Parametric Positioning and Adrawings

Understanding Parametric Positioning

Both the Parametric Assembly and Explicit Assembly environments enable you to explicitly position components (specifically their associated model geometry) in 3D space using tools such as Move Component and the View and Position option of View Component. However, the Parametric Assembly environment enables you to parametrically constrain components using the following:

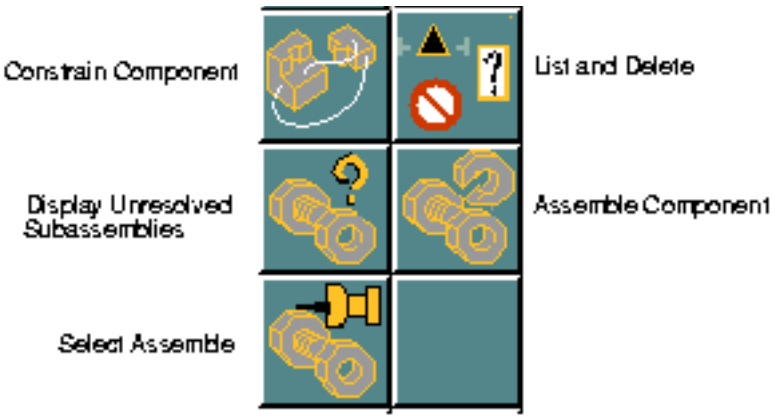
- Assemble Component
- Constrain Component
- Select Assemble

This provides more powerful design possibilities and controls than simple explicit positioning of the components.

Please note: Interpart constraints, described in Chapter 9, “Using Interpart Constraints”, and parametric positioning, described in this chapter, are two separate and distinct means of using parametrization in an assembly. They are two different systems.

Task Set Options

The parametric positioning icons are shown below:



Constrain Component: Enables you to create parametric positioning constraints in the active main assembly as well as in the active reference assembly. See “Using Constrain Component” on page 6-13.

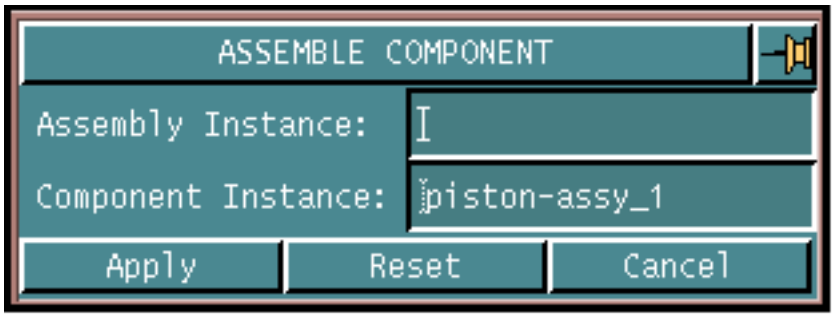
Modify Position Constraints: Enables you to view all the positional constraints in the selected subassembly. You can modify or delete these listed constraints. See “Modify Position Constraints:” on page 6-17.

Display Unresolved Subassemblies: This opens the Unresolved Sub Assemblies menu. This menu shows subassemblies that have not yet been solved because of either of the following conditions:

- You have issued Select Assemble Off and not yet issued Assemble Component.
- The subassembly is either overconstrained or underconstrained and cannot be solved in its present state.

Assemble Component: If you have issued Select Assemble Off, you can selectively assemble using this property sheet. Specify the subassembly component name that you wish to assemble and choose Apply option. The positions of the specified components are updated to satisfy the constraints.

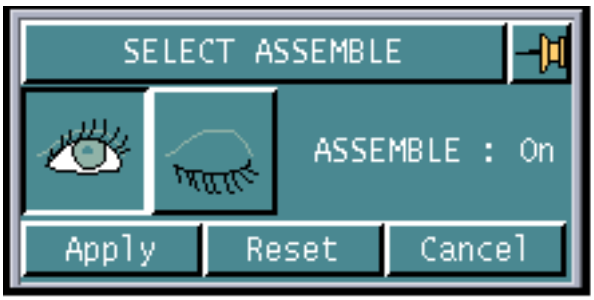
Figure 6-1 Assemble Component property sheet



For more information, see “Solving Parametric Positioning Constraints” on page 6-32.

Select Assemble: You can solve the constraints automatically or manually. For automatic update, choose Assemble On. For manual update, choose Assemble Off and then use Assemble Component to selectively solve the constraints.

Figure 6-2 Select Assemble property sheet



For more information, see “Solving Parametric Positioning Constraints” on page 6-32.

Parametric Positioning Tools

The parametric positioning capability allows you to position components by inference using positioning constraints such as mate, align, and orient. Thus, when a component moves in an assembly, the new positions of affected components can be recalculated such that the positioning constraints are satisfied.

When Does Solving Occur

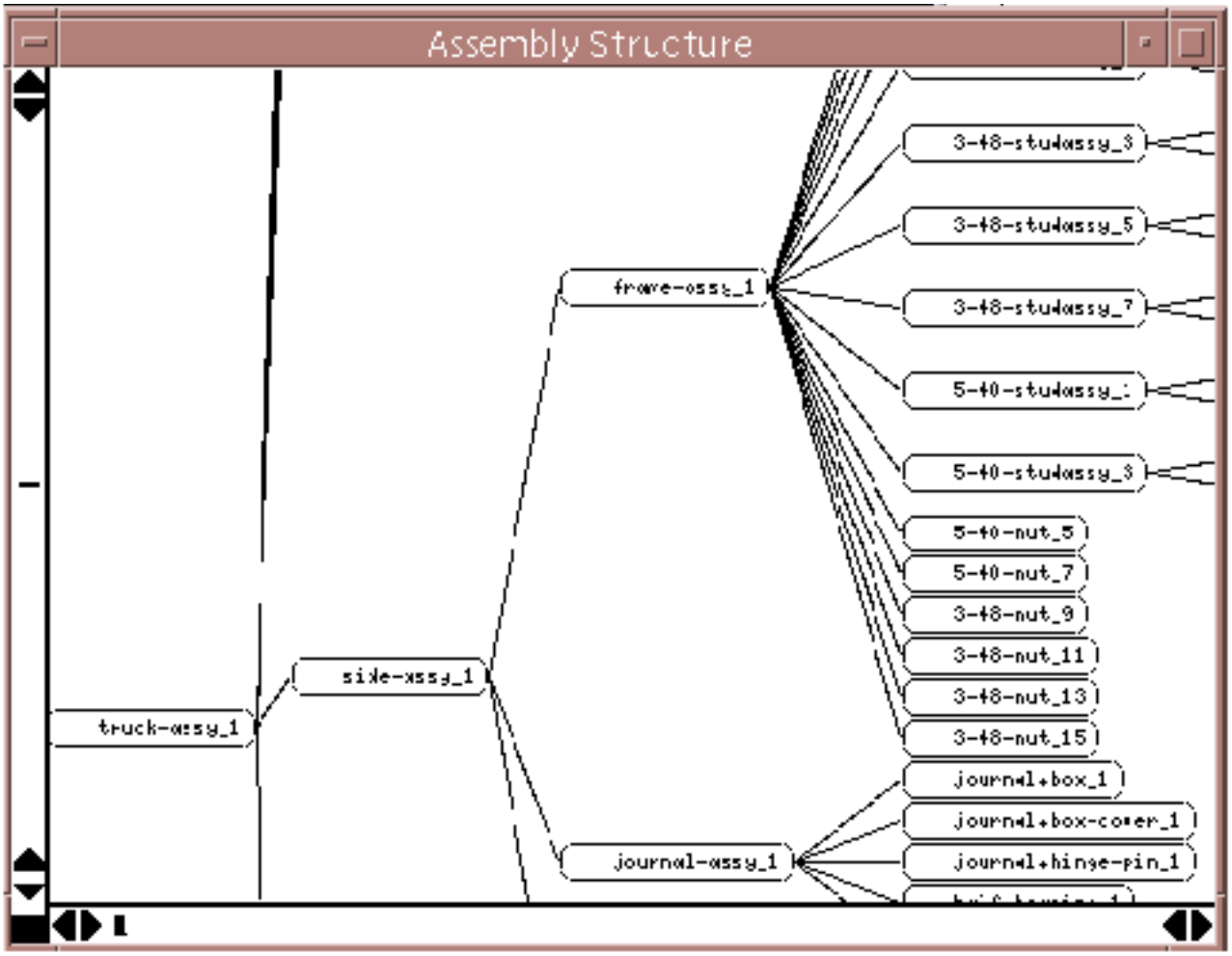
Parametric positions can be updated automatically or manually. To update automatically, first issue the Select Assemble On command. See “Automatic Assembly” on page 6-32 for detail on which commands initiate solving. Use the Assemble Component command to update parametric positions manually (see “Manual Request to Assemble” on page 6-32).

Understanding Solving Process Order

The order of solution is from right to left, or lowest level to highest level. Thus the positioning constraints are solved from the leaf components (rightmost components) to the root component (leftmost component).

In this section a truck assembly illustrates parametric positioning. The subassembly `frame-assy_1` consists of several subassemblies (such as `3-48-studassy_1, 5-40-studassy_1,` and so forth.) and components (such as `5-40-nut_5, 5-40-nut_7, 3-48-nut_9,` and so forth.). The subassembly `side-assy` consists of three subassemblies - `journal-assy_1, journal-assy_3, and frame-assy_1`.

A section of the assembly is shown below:



In the example of a truck assembly, the positions of the leaf components of the rightmost subassemblies (such as `5-40-studassy_1, 3-48-studassy_7`, and so forth.) are solved first. Then, the parametric positions related to the nearest common parent component (`frame-assy_1`) are solved. After that, subassembly `side-assy_1` is solved positioning *f*`rame-assy_1, journal-assy_1,` and `journal-assy_3` with respect to each other. Finally, the root component `truck_assy_1` is solved.

Using Positioning Constraints within Multiuser CAMU

Multiple users can work on different parts of the assembly. For example, User A might work on the `frame-assy` subassembly while User B works on the `journal-assy` subassembly, and User C works on the `side-assy` subassembly.

To specify parametric positioning constraints between the components making up `journal-assy`, User B must lock `journal-assy`. This allows him to work on the subassembly and prevents other users (such as User A) from doing so.

To constrain two leaf components, you must lock their nearest common parent.You do not need to lock the components themselves.

Please note: For more information on locking, see “Locking Components, Attribute Groups, and Models” on page 5-31. For more information on multiuser CAMU, see “Several Users Can Work Concurrently” on page 5-27.

Path Locking

Path locking facilitates creating and revising positional constraints in multiuser CAMU. If one user (User A) is defining constraints on a particular sub-assembly then other users cannot (are prevented from) change the structure of that sub-assembly until User A files (saves). Path locking is implicit behavior in the parametric positioning function. Consider the following:

Two users are working on the same assembly. User A is creating constraints on a particular sub-assembly. User B is deleting components from the assembly tree. User A creates constraints using geometry from two components. He has not yet filed these constraints. Since parametric positioning requires you to only lock the sub-assembly being constrained and not the components from which geometry is selected, it’s possible that User B may attempt to delete one of the components that User A is using as a positional constraint.

Only after User A files, User B can delete components.

If one user deletes a component that participates in a constraint, the Modify Position Constraints menu indicates that the constraint is no longer valid. For example, if User B’s actions make User A’s constraints invalid, the feedback menu (see “Modify Position Constraints:” on page 6-17) shows the invalid constraint. No other user can create a constraint using geometry from the deleted component, even if the first user has not yet filed the assembly.

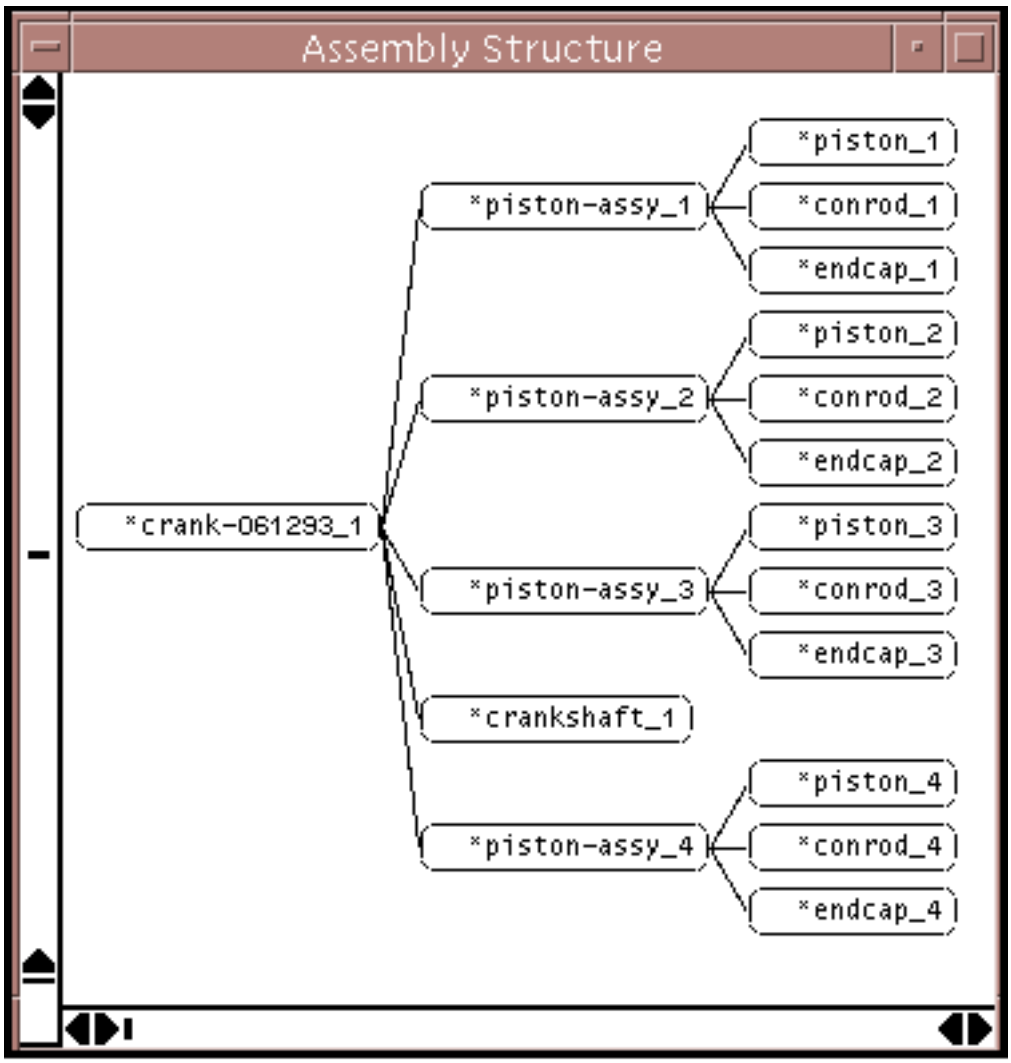
Creating Positioning Constraints

You can create positioning constraints using options such as Mate, Align, Orient, and Fix in the Constrain Component command (see “Using Constrain Component” on page 6-13). An example is provided in “Sample Procedure for Creating Constraints” on page 6-22.

Locking

You must lock the nearest common parent.

For example, in the assembly shown below, the parent component `piston-assy_1` is locked in order to specify positioning constraints between `piston_1, conrod_1,` and `endcap_1`. To specify positioning constraints between `piston-assy_2` and `piston-assy_3`, lock the root component too.



Creating Relationships between Component Geometry

You can only select components that are valid for defining the positioning constraints. For example, in the case of the crank assembly (see "Locking" on page 6-7) `piston-assy` is locked in order to add components. The `piston` and `conrod` are added. To add the `endcap` parametrically, you can select geometry only from the `piston` and the `conrod`. This is because you are positioning the components within subassembly `piston-assy`.

If the design calls for positioning the `endcap` with respect to the `crankshaft`, this implies that subassembly `piston-assy` needs to be positioned parametrically in relation to the `crankshaft`. Lock `piston-assy`, `crankshaft,` and the root component. Then define positioning constraints between the `endcap` and the `crankshaft`.

Constraint Mapping between Cylindrical Surfaces

Using the Mate option, (see "Mate:" on page 6-15) the constraint maps to tangent constraint. The two cylinders will have a line contact if their radii are different and one of them is a hole.

Using the Align option, (see "Align:" on page 6-15) the constraint maps concentrically. If you have another mating constraint between planar surfaces, and the system is over constrained, try flipping while defining an align constraint between cylindrical surfaces. Use the on-screen arrows to determine correct direction.

Ambiguous Selection

When you attempt to select geometry, the ambiguous selection mechanism cycles through all possible entities by highlighting them at the point of selection. It highlights the component in a different color in addition to highlighting the selected entity.

Using the Flip Option

Whenever an edge is selected for the purpose of parametric positioning, an arrow appears to indicate the axial direction. An option to flip direction is provided. This eliminates any ambiguity while positioning the selected components.

Entity Selection

You must select the correct type of entity for each option. For example, when you choose the Plane option, you can only select planar faces. When you choose the Edge option, you can only select straight line edges. When you choose the RevSurface option, you can only select cylindrical surfaces. When you choose the Point option, you can only select points from the model instance.

Entity selection not only depends on the first entity type picked but also the constraint type specified. For example, an align constraint between a line and a cylinder aligns the cylinder axis along the line.

Tables describing which constraint type allows which entity type are provided below. A dashed line in the tables indicates that, for this constraint, the geometry pair is not valid.

Table 6-1 Mate Constraint - Zero Offset

Type	Point	Plane	Line	Cylinder
Point	Coincident	Coincident	Coincident	Coincident
Plane	Coincident	Coincident	Coincident	Tangent
Line	Coincident	Coincident	---	Tangent
Cylinder	Coincident	Tangent	Tangent	Tangent

Table 6-2 Mate Constraint -Non- Zero Offset

Type	Point	Plane	Line	Cylinder
Point	Distance	Distance	Distance	Distance*
Plane	Distance	Distance	Distance	Distance*
Line	Distance	Distance	---	Distance*
Cylinder	Distance	Distance	Distance	Distance*

Distance* = offset + radius of the cylinder(s)

Table 6-3 Align Constraint - Zero Offset

Type	Point	Plane	Line	Cylinder
Point	---	---	---	---
Plane	---	Coincident	---	---
Line	---	---	Coincident	Concentric
Cylinder	---	---	Concentric	Concentric

Table 6-4 Align Constraint - Non-Zero Offset

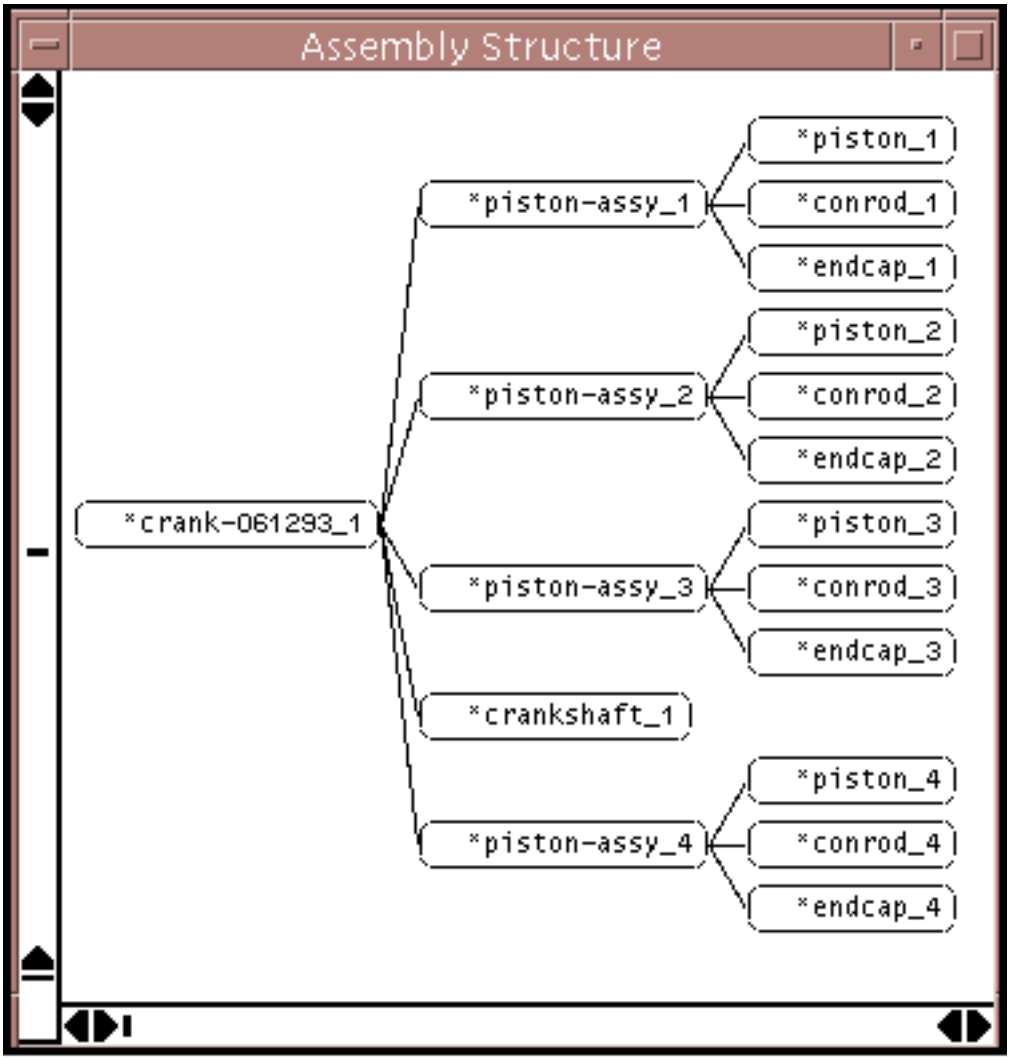
Type	Point	Plane	Line	Cylinder
Point	---	---	---	---
Plane	---	Distance	---	---

Table 6-4 Align Constraint - Non-Zero Offset

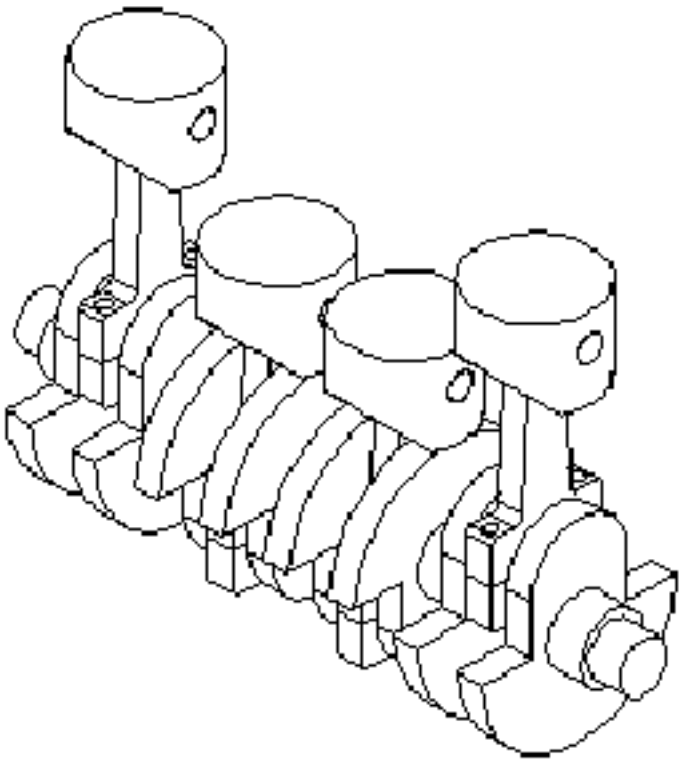
Type	Point	Plane	Line	Cylinder
Line	---	---	Distance Parallel	Distance Parallel
Cylinder	---	---	Distance Parallel	Distance Parallel

Multiple Instances of a Subassembly Class

The assembly structure shown below consists of the root node, four instances of the `piston-assy` subassembly, and one `crankshaft` component.



You cannot establish different positioning constraints between the child components of each `piston-assy` instance. The positioning constraints between the components of any `piston-assy` subassembly are the same for all instances of that subassembly. The relative orientation (relationship to one another) of `piston, conrod`, and `endcap` is the same for all the instances of `piston-assy`.



Creating a New Class to Capture Unique Positioning

If you want to change the positioning constraint of any instance in a multi-instance subassembly class, for example to orient `piston_1` to `conrod_1` at a different angle than all other piston/conrod relationships in `piston-assy`, you must create a new class, such as `piston-assy-15degrees`. If you do not create the new class manually, CAMU will create it automatically if you use the Instance Only option on the property sheet.

If you change class name using the Change Component property sheet, you can use the Retain/Not Retain option to keep or discard existing constraints for the new class name.

Please note: For more information on how to create a new class rather than propagate a change to all instances of a class, see "Changing a Component Name Can Create a New Class" on page 7-6.

Hybrid Positioning

You can explicitly and parametrically position components in existing and new assemblies.

Parametric and Explicit Positioning

Any assembly can have a combination of explicitly positioned components and parametrically positioned components.

An explicitly positioned component can be repositioned parametrically by using parametric positioning functionality. Similarly, any parametrically positioned component can be repositioned explicitly by deleting the associated constraints. Its location will then no longer be driven by changes made to adjacent components. Laying out a 3D assembly and experimenting with different "what-if" configurations is a key strength of CAMU.

Explicit Models

If explicit models that contain geometry used to define positioning constraints are modified, you may again need to specify constraints. For example, during Explicit Boolean operations or edge filleting, the original entities (faces, edges, and vertices) are deleted. You must recreate these constraints. When this occurs, the Modify Position Constraints menu displays a No Geometry status.

Using Reference Assemblies

The assembly instance (root node of a reference assembly) is treated as a fully constrained subassembly. Its orientation and position can be specified or changed as a group (single unit). It is not possible to specify or change positional constraints between its leaf components.

You can create constraints between geometries inside a reference assembly and another reference assembly or main assembly.

Using Constrain Component

Constrain Component enables you to create parametric positioning constraints in the active main assembly as well as in the active reference assembly. Constraint options are Mate, Align, Orient, and Fix. You can list and optionally delete constraints for the selected parent component. You can also select a base component (see "Designating a Base Component" on page 6-34).

See "Sample Procedure for Creating Constraints" on page 6-22, for a graphic example of constraint creation.

If you have set Select Assemble option ON (see "Automatic Assembly" on page 6-32) all parametric positioning constraints are solved when you constrain a component.

Please note: Each command line defines constraints for a single subassembly (identified by the specified CINAME). Do not specify another CINAME (component instance name) in the same command line.

The following properties are applicable when Select Assemble option is set ON.

After you have selected the modifier on the property sheet (mate, align, or orient), you can select one or more pair of entities. Look for prompts in the Command window. After specifying the first pair, wait to see the result. Your options are:

- If you are satisfied with the result, select another pair of entities or another modifier.
- If nothing happens when you select the first pair, you can use the Modify Position Constraints option to delete undesired constraints or select other options without exiting the command.
- If the result is not satisfactory or if components visually move out of view, undo the operation by pressing the backspace key.

  Please note: Press the backspace key until the asterisk representing the entity selection is erased from the command line.

  - Pressing the backspace key after the second pick undoes the last created constraint and moves the component to the position it was at before you created that constraint. You can press backspace key once again to unselect the second entity. You can now reselect the second entity or press the backspace key again to undo the first pick. You can then reselect the first entity and then the second entity.
  - Pressing the backspace key after the first pick undoes its selection. You can now reselect the first entity or specify another modifier (mate, align, or orient).

Please note: You can only use the backspace key to undo selection of the last pair of entities. Further backspacing does not undo the constraints created earlier in the command.

Aborting the command undoes the creation (and solution) of the last constraint created in the current command line and ends the command.

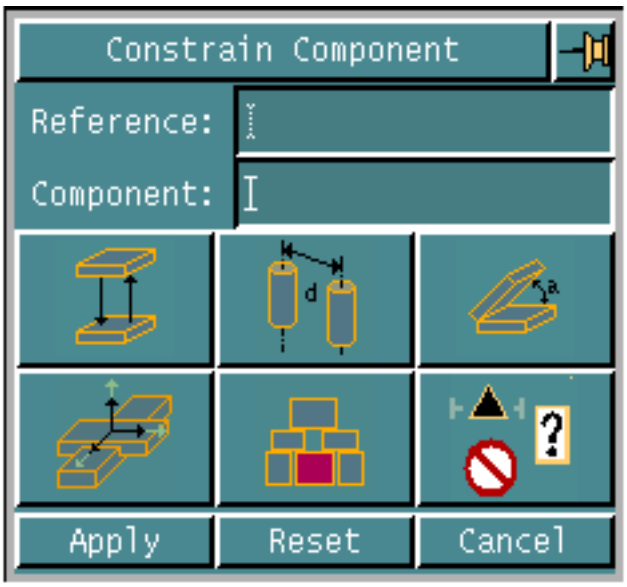
Applying the property sheet exits the command. If you have exited the command and are not satisfied with the result, undo the Constrain Component command using the CAMU Undo (UNDO ACOMMAND).

Please note: Deleting constraints does not reposition components. To reposition components while still active in the command, use the backspace key as just described. You can also abort the command. To reposition components after you have exited the command, use the CAMU Undo as just described.

Constrain Component Property Sheet Options

Select the Constrain Component icon from the task set to open the Constrain Component property sheet.

Figure 6-3 Constrain Component property sheet



Reference: Specifies the active reference assembly whose node you want to positionally constrain.

Component: Specifies the subassembly (component instance name) whose child components you want to positionally constrain. All the child components are highlighted indicating the subassembly graphically.

The Constrain Component property sheet processes your inputs real time. Only use the Apply option to exit the command. For an example of usage, see “Sample Procedure for Creating Constraints” on page 6-22.

Component Name: Specifies the subassembly (component instance name) whose child components will be constrained. To pass input to this field, select the component instance from the assembly structure using the left mouse button or enter the component name in the Component field.

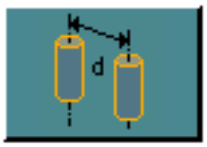
Mate:

Use the Mate option to define mating constraints between faces of components being assembled. The faces can be planes or surfaces of revolution. For planar faces, the command enforces the constraint that the planes will be parallel to each other and face each other (their normals will be in opposite direction). You can specify an offset such that the distance between the faces will be maintained equal to this offset. If no offset is specified, the faces will be coincident.

For revolved surfaces, the Mate option ensures that the two faces touch each other at a point or along a line. The mating surfaces face each other (the normals are in the opposite direction).

The two faces that mate are identified by selecting the edges of the face. For the plane option, the faces must be planar. You can specify an offset value between the mating faces.

Align:



Use the Align option to align points, the normal of planar faces, edges of solids, and/or axes of revolved surfaces. When you align points, two points on the mating components are made to coincide. Specify the two points by selecting them from the associated model of the applicable components.

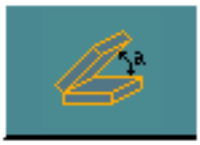
You can also use this option to align two planes such that the planes remain parallel at a given offset and face the same direction (the normal to the face is in the same direction). If the offset value is zero, the planes coincide. Similarly, you can specify that the edges of two solids be aligned, or made parallel to one another.

You can use this option to make faces coaxial if the selected faces are revolved surfaces of cylinders.

Acceptable methods and inputs are:

- Plane-Plane: Two planar faces from mating components, along with an offset value (default offset value is zero)
- Edge-Edge: Straight line edges of solids
- RevSurface: A cylindrical face
- Align: Between selected entities such as lines and planes

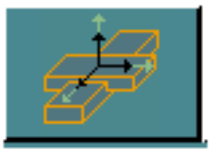
Orient:



Use the Orient option to orient two planar faces of mating components to be at a specified angle with respect to each other. In case no offset value is specified, the distance between the faces is an unspecified degree of freedom.

The input for this option is the two faces of the mating components and a value for the angle between them. You can orient planar faces and edges using this option.

Fix:



Use the Fix option to assemble a pair of components by superimposing coordinate systems attached to each component. By using this option, the two components are fully constrained.

Please note: Both models must contain the identical CPLane (coordinate system). Use the Define CPlane command (while active in each model) to create them or use one of the seven supplied CPlanes.

Inputs are the two CPLs as stated above.

The user interaction sequence is:

1. Select the first component. Its associated model highlights in the graphics window. All CPLanes in the associated model appear in the runtime list.
2. Select the desired CPlane name from the runtime list.
3. Select the second component. Its associated model highlights in the graphics window. All CPLanes in the associated model appear in the runtime list.
4. Select the same CPLane name for the second model as you did for the first.

Please note: If you define an align constraint on a pair of directed entities (lines, planes, cylinders), and want to flip the direction of one of the entities, use the flip option on the utilities pop up. There is no flip option on the Constrain Component property sheet.

Base Component:

Use the Base Component option to name a base component. The base component is the one with respect to which all other components are positioned. (For details, see “Base Component” on page 6-33).

Modify Position Constraints:

The Modify Position Constraints option allows you to view all the positional constraints in the selected subassembly. It displays a runtime list of previously defined positional constraints for the specified component instance.

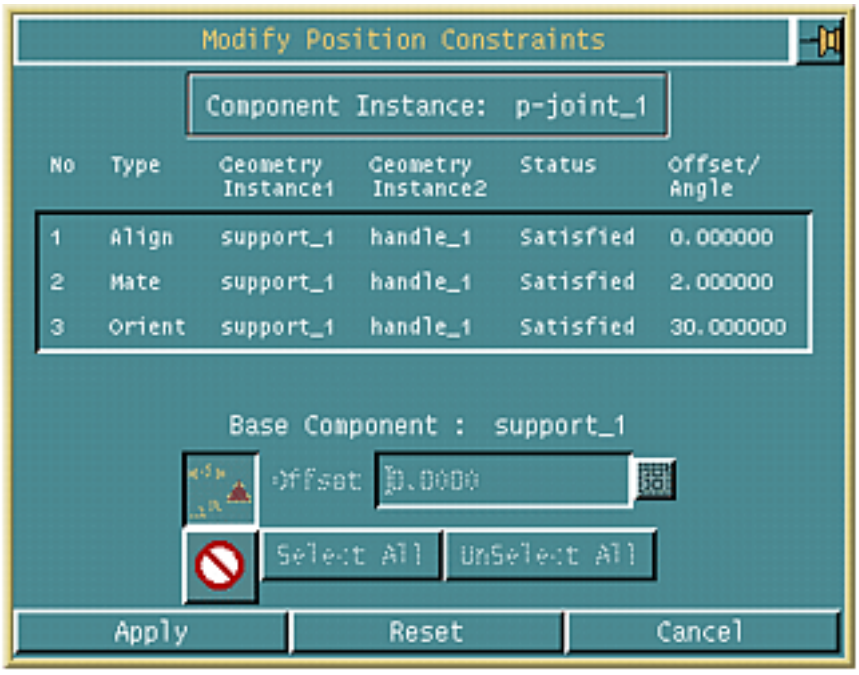
You must lock the parent subassembly node whose children components are positionally constrained.

You can use this option to:

- Get a List the existing constraints
- See the status of the constraints
- Change Offset value of a selected constraint
- Delete selected constraint(s).

Please note: The runtime list on this menu updates in real time as you create constraints

Figure 6-4 Modify Position Constraints Menu



The name of the selected component instance is shown in the Component Instance field.

List: You can view a list of existing position constraints of the specified component. The list gives details of type of constraint, geometry instance, status and offset values of the constraints.

You can also get the listing to a report file by setting the environment variable AD_LIST_CONSTRAINTS to “yes” in your `.caddsrc` or `.caddsrc-local` file.

```
setenv AD_LIST_CONSTRAINTS "yes"
```

Status: Following are the various status of the listed constraints:

Invalid Constraint: The No Geometry message in the Modify Position Constraints menu could mean any of the following:

- The model geometry referenced by the constraint has been deleted.
- The component to which one of the geometry belongs has been deleted.
- One or both of the referenced components no longer have the original common parent.
- The model associated with one or both of the components has been changed.

Under Constrained: This indicates that the geometry within that subassembly cannot be uniquely positioned due to insufficient constraints.

Fully Constrained: This means that the assembly is well defined and all degrees of freedoms have been exhausted.

Overconstrained: This indicates that the dimension system is overconstrained or is in a configuration that the solver cannot handle. It implies that one or more constraints need to be removed.

Cannot Solve: This generally indicates that the data is inconsistent. To resolve, you should delete some constraints and recreate them with different values. You may also see this message if the data is underconstrained and the solver has chosen defaults which do not lead to a solution. In this case adding a few more constrains may help resolve the problem.

Not Changed : This indicates that component positions have not been moved due to conflicts with other components or constraints that may be overconstrained or that the solver could not solve.

Satisfied: This indicates that the constraint is solved completely.

Changing Offset Value

You can change the offset value of a selected constraint, available from the runtime list. To do so, proceed as follows:

1. Click on a constraint from the available runtime list, of which you want to change the offset value.
2. Specify the new offset value in the Offset field, using calculator.
   - If the selected constraint is of type Mate or Align, the value will be in drawing units.
   - If the selected constraint is of type Orient, the value will be angle in degrees.
3. Select Apply option to change the offset value of the selected constraint.

The model geometry associated with selected constraint is highlighted according to the condition of the constraint.

Deleting a Constraint

The Delete options on the Modify Position Constraints property sheet enables you to delete any existing constraint. Once you select the Delete option you can select multiple items from the runtime list and delete them all at once.

1. Click on the Delete icon.
2. Select the constraint(s).
3. Select Apply option on the property sheet and exit the command.

   The parametric positioning option used to define that constraint is removed from the database.

Please note: Once the constraint is deleted, it cannot be undeleted.

Accessing Modify Position Constraints Menu

You can open the Modify Position Constraints menu in one of the following ways:

- When you activate or refresh the assembly or when you issue the Assemble Component command, than select the Unsolved Sub-assembly option. If you select a subassembly the Modify Position Constraints menu appears.

  If you open the menu in this manner, the runtime list displays only unsolved, overconstrained, and underconstrained constraints. It will not display the solved ones.
- When you issue the Constrain Component command, and select the Modify Position Constraints option. It shows all the constraints defined for the selected subassembly.
- You can select the Modify Position Constraints option from the COMP/ASSY task set.

Example: Deleting Invalid Constraints

In this scenario, the `piston-assy` has been created by placing a mate and an align constrain between the `endcap` and the `conrod`.

1. You delete some or all of the underlying geometry used to define a constrain or constraints.

   This makes the affected constraints invalid.
2. You file or refresh the assembly or attempt to solve the invalid constraints.

   This opens the Unsolved Sub Assemblies menu.

3. You select the `piston-assy` from the list.

   The Modify Position Constraints menu appears.

4. You can delete the invalid constraints now.

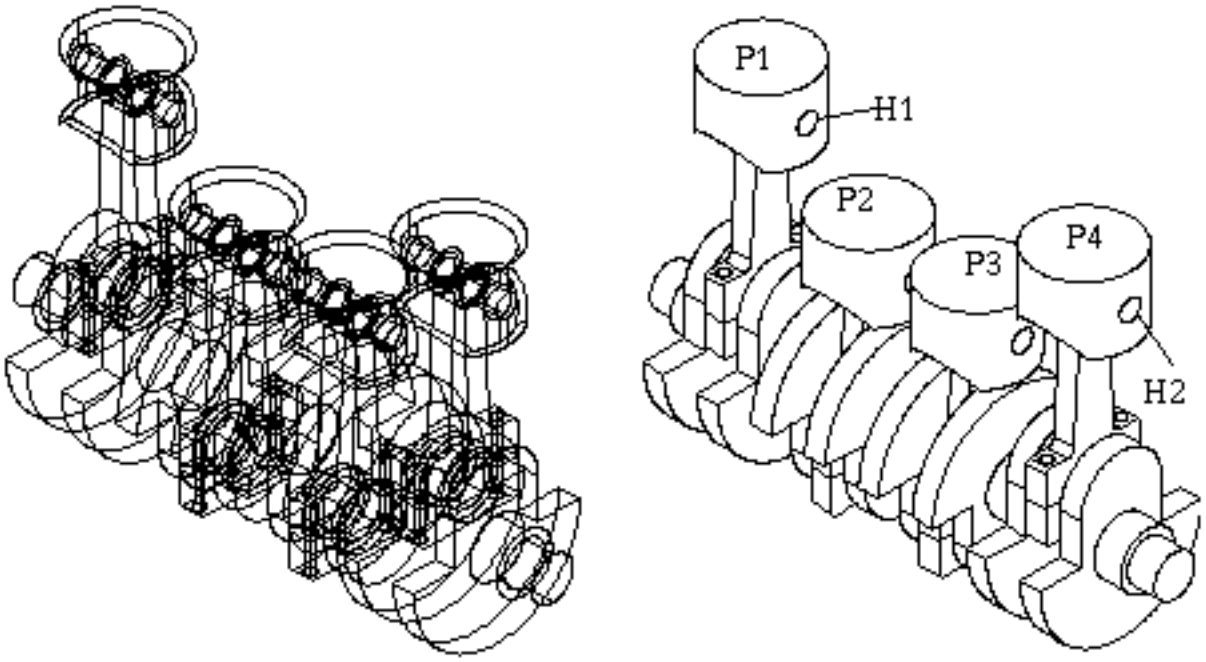
Color Scheme

CAMU uses colors to indicate the status of positional constraints. You can modify the color scheme using an environment variable in your `.caddsrc-local` file (see ".caddsrc-local File Information" on page 2-3).

Example: Isolate Constraint Problems

In this scenario, a set of constraints inconsistently overconstrain the assembly. You must delete several constraints to solve this problem.

Consider the crankshaft assembly shown below:



You have created three align constraints as follows:

- Between geometries P1 and P4 belonging to `piston-assy_1` and `piston-assy_4` respectively

- Between geometries P2 and P3 belonging to `piston-assy_2` and `piston-assy_3` respectively
- Between H1 and H2 belonging to `piston-assy_1` and `piston-assy_4` respectively

The constraint between holes H1 and H2 is an align with an offset value. The other two constraints have offset zero.

The offset value between H1 and H2 is incorrect. The assembly will not solve because this error creates an inconsistently overconstrained assembly.

1. Solve the constraint.

   The following output appears in the Modify Position Constraints menu:

Align	piston-assy-1	piston-assy-4	Overconstrained
Align	piston-assy-2	piston-assy-3	Overconstrained
Align	piston-assy-1	piston-assy-4	Overconstrained

2. Select the assembly from the runtime list.
3. Select any of the constraints in the list to see the underlying geometry highlight in the graphics window.

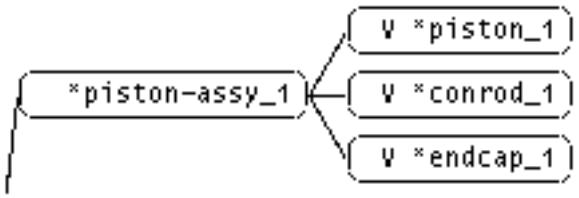
Please note: Highlighting helps you distinguish between constraints and see constrained geometry pairs. You can easily see and delete unwanted or redundant constraints.

4. Resolve the conflict and solve the system either by correcting the offset or removing the align constraint between P1 and P4.

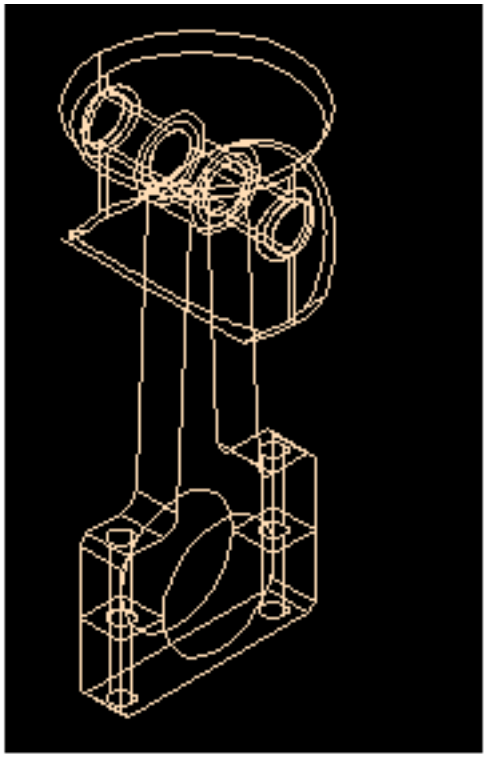
Sample Procedure for Creating Constraints

In this example, we will create a mating constraint between the connecting rod and endcap in the piston subassembly.

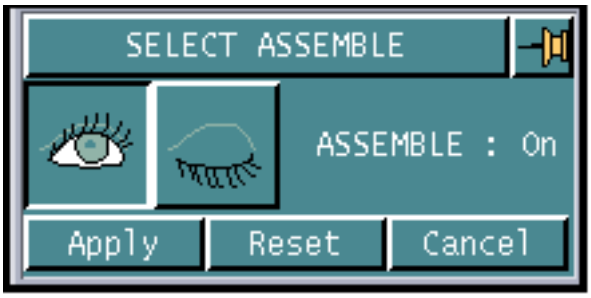
1. Lock all affected components.
2. Display one of the piston subassemblies in the sample crankshaft assembly.



3. Manipulate view display to scale and position the associated model geometry in the Graphics window.

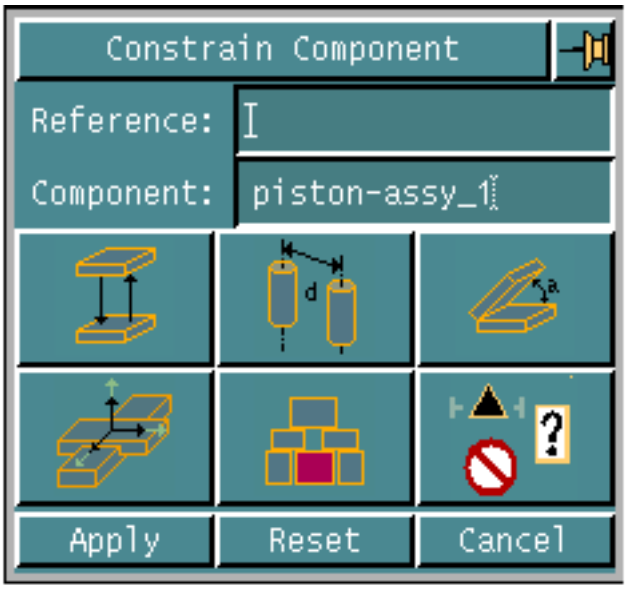


4. Issue the Select Assemble On command.

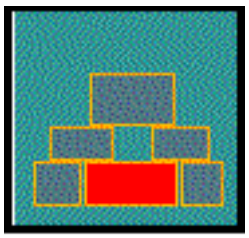


Please note: If you issue the Select Assemble On command at this stage, the constraint are solved immediately on satisfying the specific constraint. Constraint solving is delayed until you select the Select Assemble On.

5. Open the Constrain Component property sheet.



6. Specify the name of the parent component. Using the left mouse button, select the `piston-assy_1` component in the assembly structure. This fills in the name in the Component name field.
7. Select a base component. Select the Base Component icon from the property sheet.



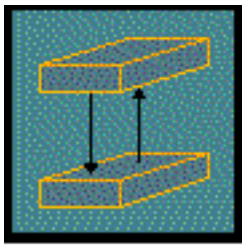
The property sheet updates to display the Base Component name field.

Please note: Specifying a base component is optional, but recommended. If you do not specify one using the menu, the system defaults to the first component that you select in the Constrain Component command. A subassembly can only have one base component.

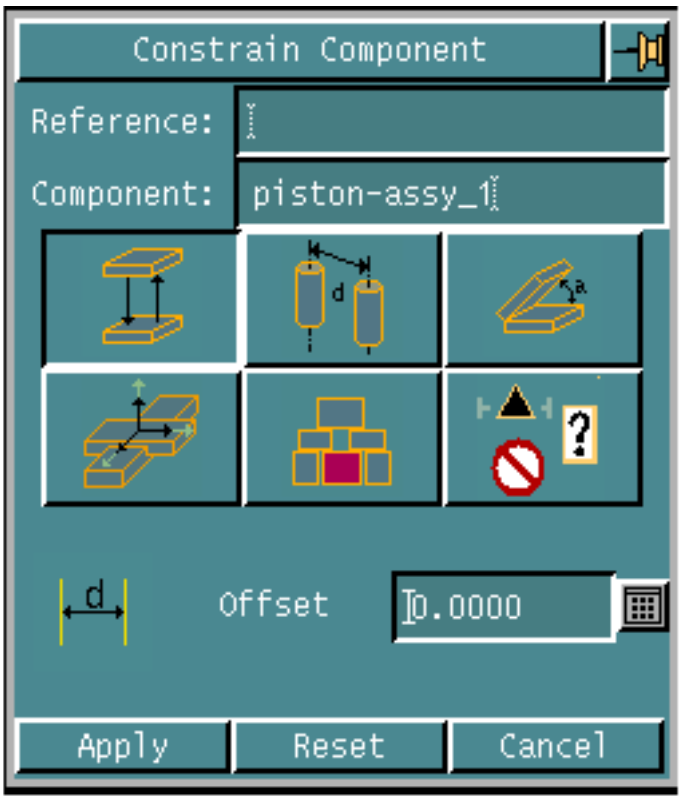
8. Using the middle mouse button, select either `endcap_1` or `conrod_1` as the base component. In this example, we will select the connecting rod.

The system immediately processes your request and recognizes `conrod_1` as the base component for the `piston-assy_1` subassembly.

9. Select the Mate option from the Constrain Component property sheet.



The property sheet updates as follows:



The following two lines appear in the Command window:

```
Select two geometries and apply.
Constrain Component Ciname piston-assy_1 Mate
```

10. Specify an Offset distance of 0.

Use the calculator to specify the offset distance value. The default value is 0.

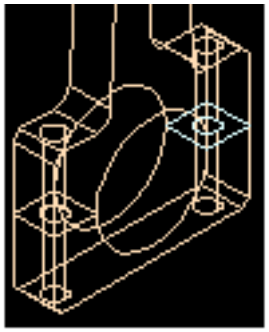
Please note: You cannot specify a negative value for the offset distance.

11. The command line immediately updates as follows:

```
Constrain Component Ciname piston-assy_1 Mate Offset 0
```

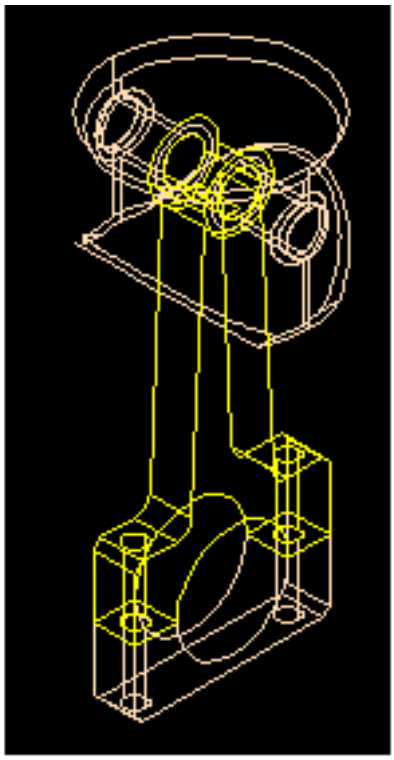
You can now select pairs of geometry to mate and align. In this example, specify a mating alignment using a single pair of viewed entities.

12. Select the shared face between the connecting rod and endcap.

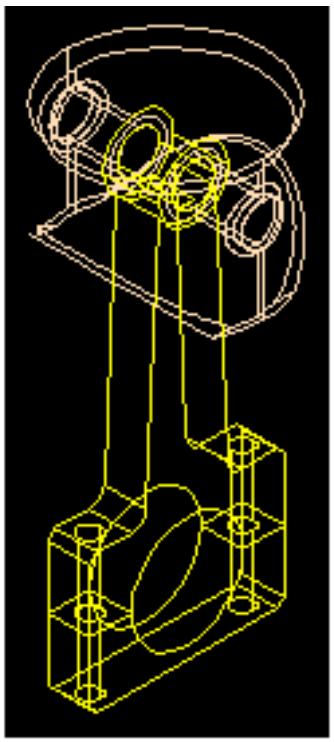


13. Use the Ambiguous menu to select either the connecting rod or endcap face.

In this example, the connecting rod is specified first.



14. Select the second component. Use the Ambiguous menu to select the endcap face. The graphics update as follows.



Please note: It is difficult to constrain already assembled components since undesired entities are highlighted. This happens when a line coincides with another. To avoid this situation, separate the components prior to constraining.

Please note: The two components are now mated, at the specified face, at a distance of 0. If you move one of the components, the other will move to satisfy the constraint.

The command line now looks like this:

```
Constrain Component Ciname piston-assy_1 Mate Offset 0 * *
```

Please note: If you are dissatisfied after selecting the first or second component, use the backspace key to undo the selections. See "Using Constrain Component" on page 6-13.

15. Click Apply on the Constrain Component property sheet to solve the constraint and exit the command.

Please note: You can select another pair of component edges or faces, specify a different offset value, or specify a new constraint type.

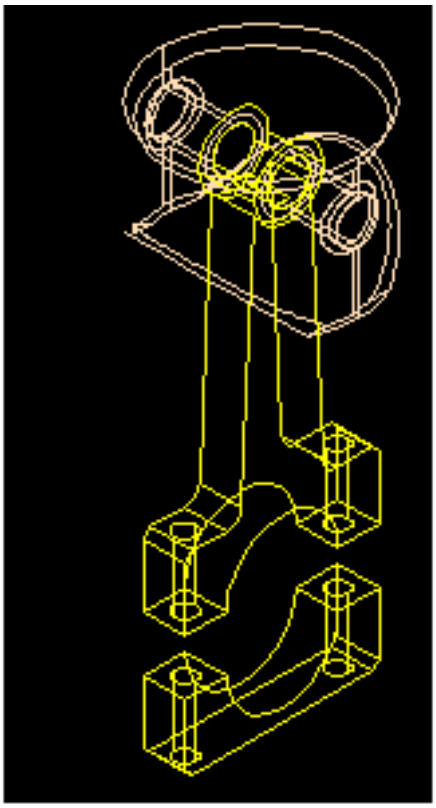
Example Variant 1

If you had specified an offset value, the endcap would have moved the specified value as soon as it was selected. For example:

1. Specify an offset of 1.
2. Select the connecting rod.
3. Select the endcap.

```
Constrain Component Ciname piston-assy_1 Mate Offset 1 * *
```

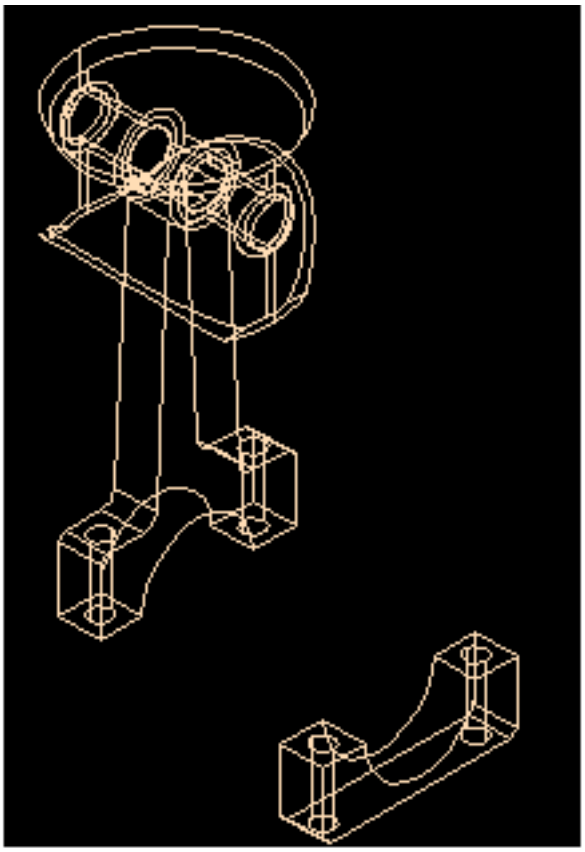
The second component moves immediately to satisfy the constraint.



When you move either the endcap or the connecting rod, the mating component updates.

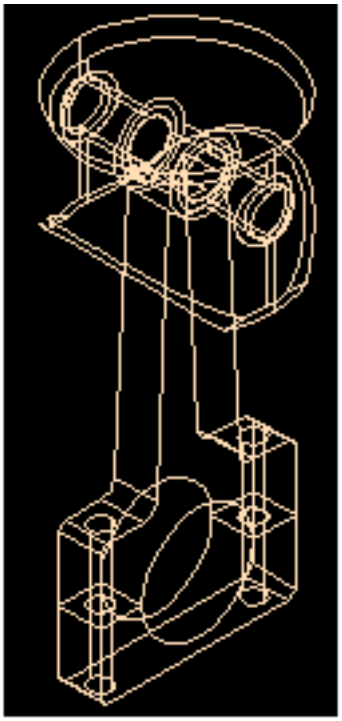
Example Variant 2

If the endcap and conrod had not been properly positioned prior to creating the constraint, you could have properly positioned and constrained them at the same time. For example, the components are originally positioned as follows:



Repeat the steps in “Sample Procedure for Creating Constraints” on page 6-22.

The endcap moves as follows:



Solving Parametric Positioning Constraints

You can solve parametric positioning constraints automatically or manually.

See “Understanding Solving Process Order” on page 6-4 for more information about how the system solves parametric positioning constraints.

Automatic Assembly

The Select Assemble On command enables you to specify that all parametric positions be recalculated and solved (assembled) whenever the following are executed:

Command	Icon or Property Sheet
Activate Assembly	LDM
Add Assembly	Add Assembly
Copy Component	Copy Component
Constrain Component	Constrain Component
Orient Component	Move Component
Position Component	Move Component
Refresh Assembly	side bar Refresh icon
Rotate Component	Move Component
Translate Component	Move Component

This method is suggested.

Manual Request to Assemble

In case you have used the Select Assemble Off command, use the Assemble Component command to assemble all the child components of the selected parent component.

The components are assembled by solving their parametric positioning constraints, and are placed at their newly computed positions/orientations.

Please note: A subassembly can be assembled without locking anything.

Base Component

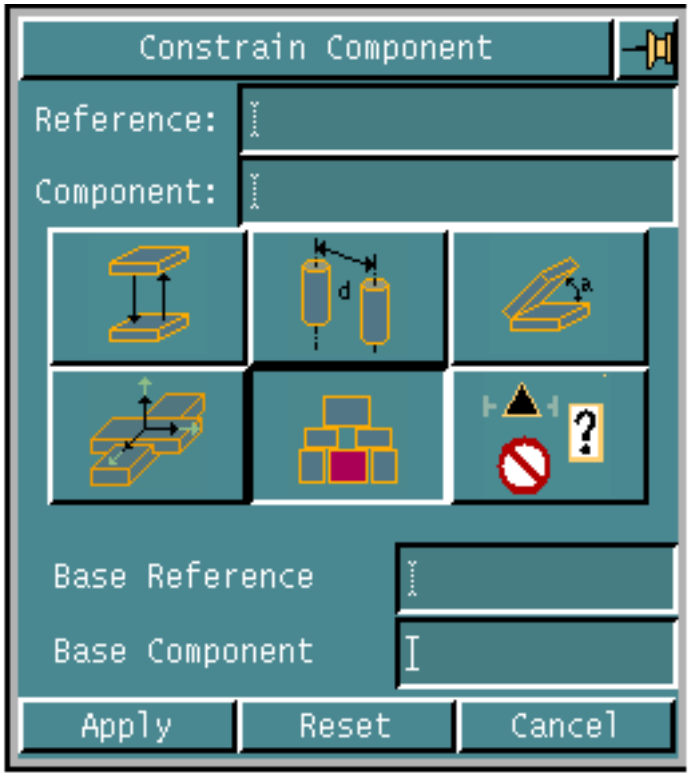
The base component is the one with respect to which all other components are positioned. It is used as a datum in order to assemble the children of the selected parent component together. It remains fixed when the constraints are solved. There can be only one base component for each subassembly for which parametric positioning has been defined.

Base component behavior is described below:

- Each component (subassembly) can have only one base component.
- A base component is specific to a subassembly.
- You can select the base component from the main assembly as well as from a reference assembly.
- You can select the base component by either choosing the component graphically or from the assembly tree structure.
- A base component is specified as the CINAME of any of the child nodes of the subassembly on which constraints are defined.
- Internally the base component is mapped to the parent relation of the child subassembly. This makes the base component instance independent.
- If you do not specify a base component while creating constraints, the instance from which the first geometry is selected becomes the base component.
    - Even if there is a model associated with the owner component, it will not be taken as the base component for that subassembly.
    - You can change a base component designation.
- If all the constraints specified do not map to the base component, a warning is issued.

On selecting the Base Component icon from the Constrain Component property sheet, the options specific to the base component appear on the property sheet.

Figure 6-5 Constrain Component property sheet with Base Component details



- Reference: Specifies the active reference assembly whose node you want to positionally constrain.
- Component: Specifies the subassembly (component instance name) whose child components you want to positionally constrain. All the child components are highlighted indicating the subassembly graphically.
- Base Reference: Specifies the reference assembly that contains the base component.
- Base Component: Specifies the component from the main assembly or from the active reference assembly as a base component.

Designating a Base Component

You can identify the base component yourself or let the system establish one by default. To identify the base component by name, use the Constrain Component command (see “Using Constrain Component” on page 6-13). The default base component is the one whose geometry you identify (select) first when creating a parametric positioning constraint.

Redesignating a Base Component

You can redesignate the base component at any time using the Constrain Component command and specifying a new base component.

Recognizing Full Constraints Status

An assembly that contains parametric positioning constraints is understood to be either fully constrained, underconstrained, or overconstrained.

When you issue the Assemble Component command, the positions of all the components for the selected subassembly are computed. At this time the system makes the determination whether the assembly is fully constrained, underconstrained, or overconstrained and acts accordingly.

Fully Constrained Assembly

As a result of the Assemble Component command, the assembly is viewed in its assembled configuration with all positioning constraints satisfied.

Underconstrained Assembly

As a result of the Assemble Component command, there will be an attempt to satisfy existing constraints.

If the specified positioning constraint set is underconstrained, an optimal solution will be provided.

Overconstrained Assembly

As a result of the Assemble Component command, there will be an attempt to satisfy constraints and eliminate redundancies.

If the positioning constraint set specified by the user is overconstrained, there will be an attempt to eliminate the redundant constraints, provided they are consistent. If they are inconsistent, the status in the Modify Position Constraints menu will be displayed.

Changing Positioning Constraints

You cannot change an existing parametric positioning constraint. You must delete and redefine the constraint (see “Modify Position Constraints:” on page 6-17 and “Using Constrain Component” on page 6-13).

Updating Positions After Modeling Change

Parametrically positioned parts that have been edited at the piece part level are updated positionally when you activate the adrawing and file the suspended part. In multiuser CAMU, parametrically positioned parts are positionally updated when the other user files the changed part and you refresh the assembly.

For example, User A and User B are working concurrently in an assembly.

User A:

1. Activates a model that has been positionally constrained
2. Makes a modeling change to the parametrically positioned model
3. Activates the adrawing, thus suspending the part.
4. Files the suspended part.

   User A sees the positional effects of that change when he files the model.

User B:

1. Refreshes the assembly

   User B sees the positional effect of User A's actions.

Please note: To facilitate this scenario, you must set the following in your `.caddsrc` or `.caddsrc-local` before invoking a CAMU session:

```
setenv CADDSASSOC_UID "yes"
```

For more details see "CADDSASSOC_UID" on page 6-38.

See "Several Users Can Work Concurrently" on page 5-27 to learn more about working in a multiuser assembly environment.

Using Pre-Revision 8.0 CADDS Models

To use parametric positioning functionality on pre-revision 8.0 parts, you must do the following:

1. Set the following in your `.caddsrc` or `.caddsrc-local`:

   `setenv CADDSASSOC_UID 'yes'`

   See `/usr/apl/cadds/scripts/templates/.caddsrc-all` for description of this environment variable. This is the default setting.
2. Activate the part/model using either ACTIVATE PART or ACTIVATE MODEL in the Explicit or Explicit Assembly environment.
3. Issue the following command:

   TAG ENTITY PERMANENT ALL
4. File the part/model.

Please note: You only need to perform steps 2-4 for parts that will participate in parametric positioning constraints.

CADDSASSOC_UID

The CADDSASSOC_UID variable enables the parametric solver to retain the original task tags during model regeneration. If you have set CADDSASSOC_UID to yes in `.caddsrc-local`, the associativity of tag IDs is maintained and the solver solves constraints based on the new geometry. If you have not set CADDSASSOC_UID in the `.caddsrc-local` before regenerating a model, Refresh Assembly will attempt to solve the constraints. However, the original tags involved in the constraints will not be found. The new tags will be flagged as invalid.

Parametric Positioning and Adrawings

Parametric positional constraints are stored in the assembly database and are applicable to all Adrawings.

You can optionally issue the Select Assemble On command once for any given Adrawing. The select statement is retained when you file the Adrawing part (see “Automatic Assembly” on page 6-32).

Exploded View State

The parametric positioning capability is not available in the exploded view state. All parametric positioning functionality is hazed in the menus when you are working in the exploded view state.

Chapter 7

Changing the Assembly Structure

This chapter describes how to add components and/or assemblies to your active assembly, how to change component names and how to change their placement in the assembly structure. The topics are:

- Ways to Change the Assembly Structure
- Changing a Component
- Removing a Component
- Copying a Component
- Mirroring a Component
- Changing a Reference Assembly
- Changing Structure Appearance
- Reordering a Component
- Obtaining Information about Components
- Listing Reference Assemblies

Ways to Change the Assembly Structure

To change the assembly structure, you can change the properties of components and reference assemblies. You can also add or remove components and assemblies in your active assembly. You can change the appearance of your assembly structure by hiding portions of the structure or changing the order in which component instances appear. Select these tasks from the COMP/ASSY global task set or from the menu in the Assembly Structure window.

Changes You Can Make to the Assembly Structure

You can modify the structure of your assembly in a variety of ways.

With ADD COMPONENT you can:

- Add a new component to your assembly structure.
- Create additional instances of an existing component.

With CHANGE COMPONENT you can:

- Change the name of a component.
- Change the instance name of a component.
- Assign the component to a different parent.
- Change the model with which the component is associated, or remove the association between the model and the component.
- Associate the component with a different mode based on revision name; for example change a model association from `abx8020-a` to `abx8020-b` or remove the revision number.

With COPY COMPONENT you can:

- Copy an existing component.
- Make one or more instances of the copy.
- Give the copy(ies) a new component name.
- Assign the copy(ies) to a parent you specify.
- Have the copy(ies) associated with the same or different model.
- View and position the associated model of the copy(ies) in the context of the assembly geometry.
- Create a new component class.

With REMOVE COMPONENT you can:

- Remove a component from the structure. This also removes the association between the component and its corresponding model.

With ADD ASSEMBLY you can:

- Include an externally existing assembly in your active assembly.

With CHANGE ASSEMBLY you can:

- Change the instance name of a reference assembly.
- Change the name of the reference assembly's root component.
- Assign the reference assembly to a different parent.

With REMOVE ASSEMBLY you can:

- Remove the association of a reference assembly from your active assembly.

Changing the Appearance of the Structure

You can change the way your assembly structure appears in the assembly structure window by using REORDER COMPONENT and DISPLAY STRUCTURE.

With REORDER COMPONENT you can:

- Reorder the placement of a component instance, among its siblings, in the assembly structure.

With DISPLAY STRUCTURE you can:

- Hide (blank) portions of your assembly.
- Represent multiple instances of leaf components as a single component, with a number indicating how many instances are represented as children of the parent.

Obtaining Information about the Structure

VERIFY OBJECT gives you information about the selected component.

Changing a Component

There are several ways you can modify a component. The component must be locked for all the changes. For some changes, the parent or children must be locked. You can change one or all instances of the component.

Please note: You can use the LOCK COMPONENT command to lock a desired component. (see “Determining What to Lock” on page 5-32).

You can:

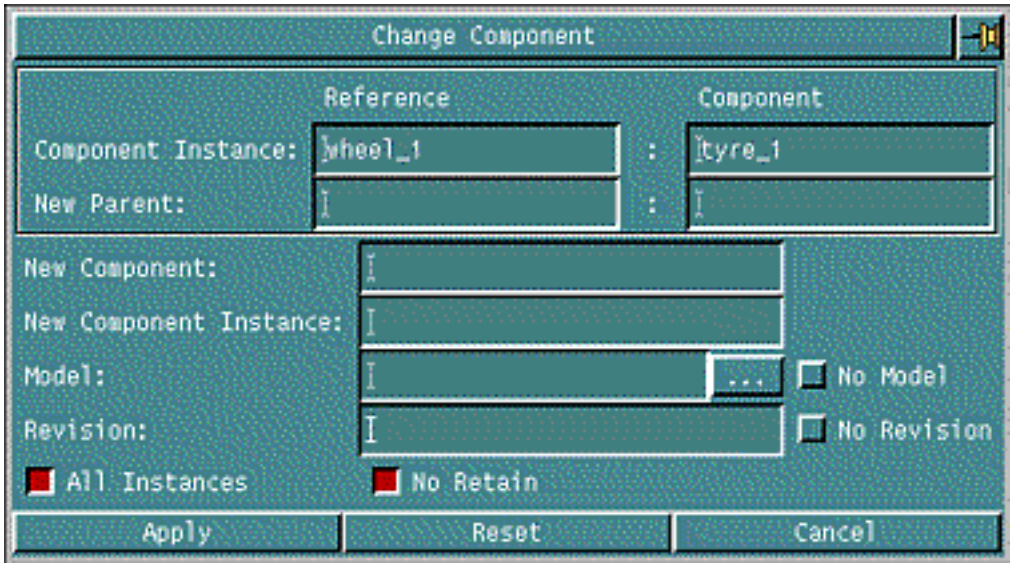
- Change the name of a component.
- Change the instance name of a component.
- Assign the component to a different parent.
- Change the model with which the component is associated, or remove the association between the model and the component.
- Associate the component to a different mode based on revision name; for example change a model association from `abx8020-a` to `abx8020-b` or remove the revision number.

Please note: If you have issued Select Assemble On (see “Automatic Assembly” on page 6-32), all parametric positioning constraints are solved when you change a component.

Steps to Change a Component

Follow these steps to change a component:

1. Choose the CHANGE COMPONENT option from the COMP/ASSY task set or select a component from the Assembly structure.
2. Choose CHANGE COMPONENT from the popup menu in the Assembly Structure window. The CHANGE COMPONENT property sheet appears.



If you are changing a component in a reference assembly, specify input in the Reference column.

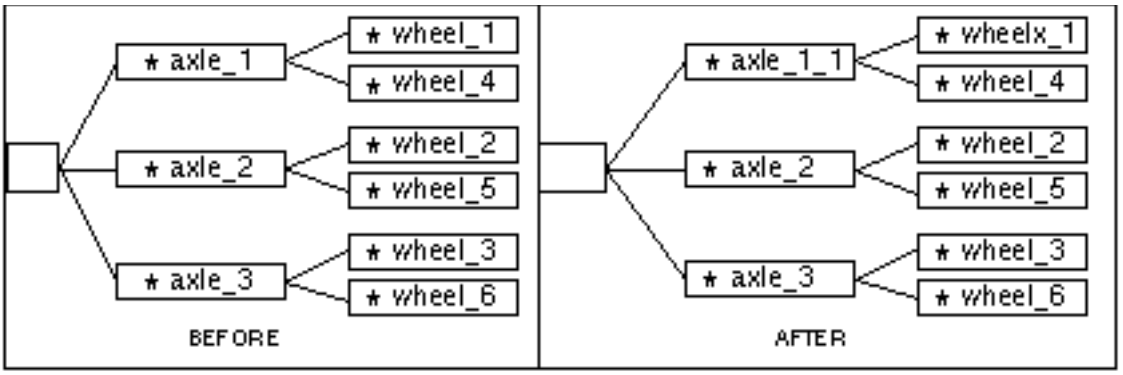
If you are changing a component in the main assembly, specify input in the Component column.

3. Enter the name of the component instance in the Component Instance field using one of the following methods:
   - Select a component from the structure using the left mouse button.
   - Select a model in the Graphics window.
   - Type the component instance name in the field.
4. Enter the new class name for the new component in the New Component field. The instance number remains the same unless you provide a New Component Instance.
5. Enter a new parent component from the assembly structure in the New Parent field.

6. Choose the No Model option to remove the association between the component from the model with which it was associated, so it has no model associated.
7. Choose the No Revision option to remove the revision number assigned to the component's associated model.
8. Choose the All Instances option to apply the change to every instance of the component class. Otherwise, the changes are applied only to the component instance you specified in the Component Instance field.
9. When creating a new component class, specify the No Retain option to either keep or discard existing parametric positioning constraints.
10. Click Apply.

Changing a Component Name Can Create a New Class

The BEFORE side of the example shows three instances of the `axle` component, each of which has two instances of `wheel`. The user changes the component name of `wheel_1` to `wheelx`, and does not select All Instances. Only one instance of the class `wheel` is changed. This forces a change to its parent component because the original `axle_1` is no longer identical to other instances in the `axle` class.



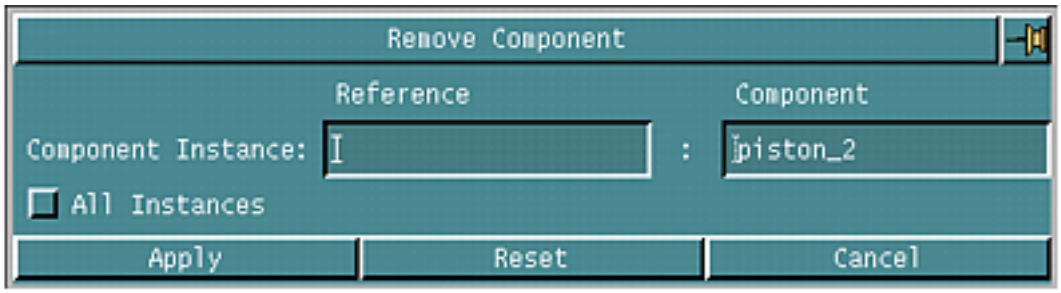
`axle_1`, `axle_2` and `axle _3` can no longer be three instances of the class `axle`, because their child components are no longer identical. Therefore, the system creates a new, unique class name, `axle_1_1`, and assigns it to the changed component's parent. The other instances of `axle` retain their original class name.

Removing a Component

When you remove a component from the assembly structure, the associated model in the Graphics window is also removed. You can remove one or all instances of a component class.

If the component you select represents a subassembly, the entire subassembly, including all child components, is removed.

The component you want to remove must be locked. Its parent component must also be locked.



If you are removing a component in a reference assembly, specify input in the Reference column.

If you are removing a component in the main assembly, specify input in the Component column.

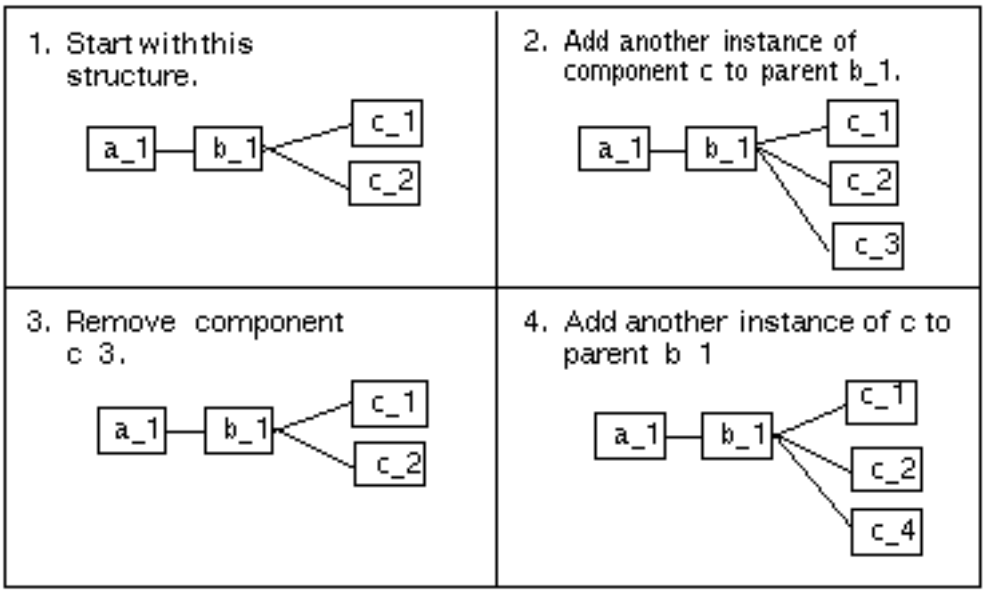
1. Choose the REMOVE COMPONENT option from the COMP/ASSY task set, or select the component from the structure.
2. Choose REMOVE COMPONENT from the popup menu in the Assembly Structure window.
3. Enter the name of the component instance you want to remove in the Component Instance field by typing, by selecting from the structure, or by selecting from the Graphics window.
4. To remove all instances of the component class, choose the All Instances option.
5. Click Apply.

Please note: The procedure describes how to remove a single component instance, or all instances of the component class, from the Assembly Structure and Graphics window. It is possible to remove selected members of a component class by selecting the relation instance instead of the component box (see "Removing a Component by Removing the Relation" on page 7-9).

Please note: If you have issued Select Assemble On (see "Automatic Assembly" on page 6-32), all parametric positioning constraints are solved when you remove a component.

Component Instance Numbers

Consecutive instance numbers are given to multiple instances of the same component class. If you remove one instance of the class, the remaining components keep their original instance numbers.



In Panel 1, the structure contains two instances of the component class `c`.

In Panel 2, the user uses ADD COMPONENT to create another instance of component c.

In Panel 3, the user removes component instance `c_3`.

In Panel 4, the user adds another instance of component `c`. This instance has the suffix `_4`, not `c_3`.

The system does not reuse the instance numbers that it has already assigned; it simply increments to the next number.

You can rename a component instance with a different number (or alphanumeric name) providing the instance number/name is not already in use.

Removing a Component by Removing the Relation

You may want to remove a component from just a specific parent class, not the entire assembly. You can do so by removing the relation instance line between the component and its parent component.

You can remove component instances by their relation in one of the two ways - All Instances or Instance Only.

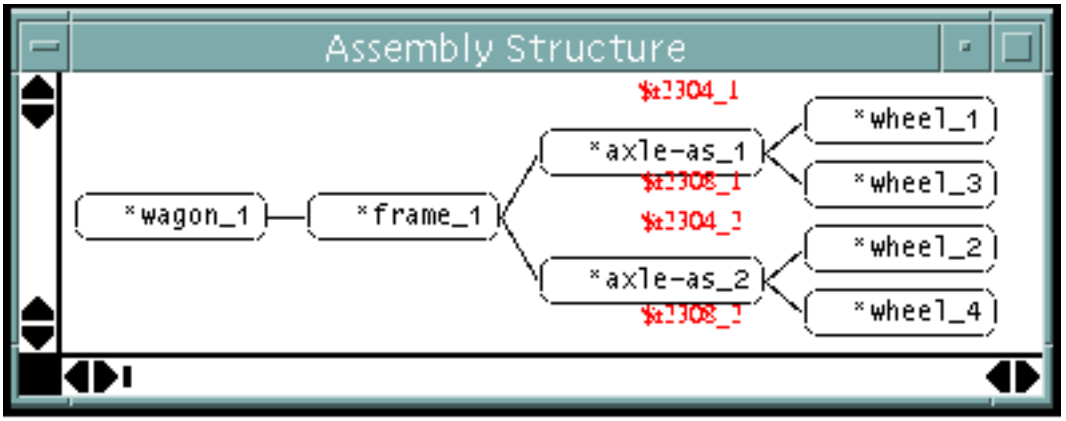
All Instances: This option removes the child component associated with the relation, from both the selected parent instance as well as all other instances of the parent class. This is illustrated in the example below.

Instance Only: This option removes just the specific child instance from the specific parent instance along the selected relation line. This action changes the component's parent to a unique class because it is now different from other members of its original class.

In this example you will remove one instance of `wheel` from each instance of `axle-as`. Specifically, you will remove the `wheel_3` and `wheel_4` instances.

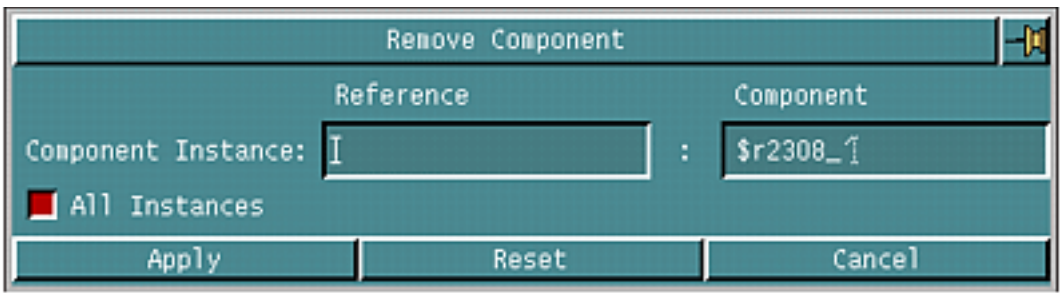
The relation between `axle-as_1` and `wheel_3` is `$r2308_1`. The relation between `axle-as_2` and `wheel_4` is `$r2308_2`. Notice that `wheel_3` and `wheel_4` are instances of the same relation class - `$r2308`.

Figure 7-1 Assembly Structure Showing Relation Instance Names

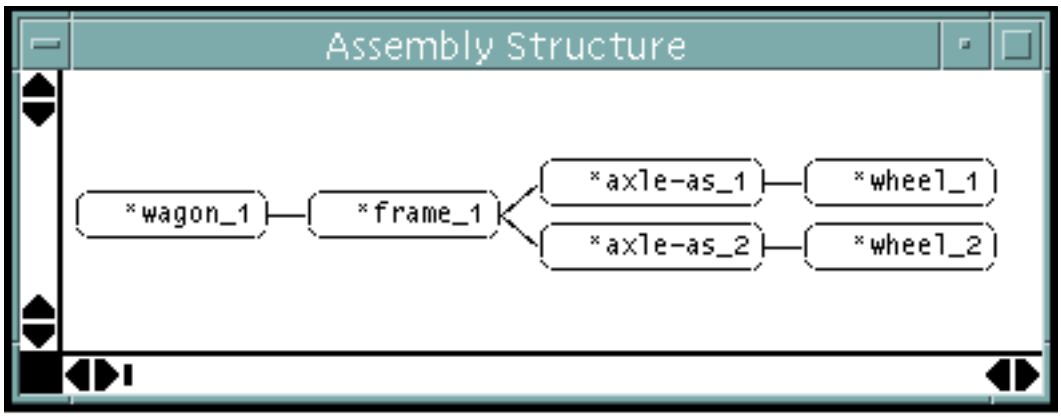


Please note: Relation names do not actually appear on the screen. Use the Detailed option on the Verify Object property sheet to see relation names (see "Obtaining Information about Components" on page 7-35).

1. Open the REMOVE COMPONENT property sheet.
2. Select the relation line between `axle-as_1` and `wheel_3` (or between `axle-as_2` and `wheel_4`). The relation value appears in the Component Instance field.
3. Choose the All Instances option to remove all instances of the relation and click Apply.



One instance of `wheel` is removed from each instance of `axle-as` along the `$r2308` relation. Associated model graphics are also removed. The names of the remaining components are unchanged.



Copying a Component

You can make one or more copies of a component and specify the parent component to which you want the copy(ies) attached.

When you copy a component, all its characteristics, including the associated attributes, if any, are copied.

The copy can have a parent different from the original component instance. If you do not specify a parent, the system attaches the copy to the original parent of the component you are copying. If you are making multiple copies, the number of copies you specify (before you click Apply) are attached to the same parent.

If the new parent has multiple instances, the copy(ies) that you make are attached to every instance of that new parent. You can give the copy a new class name. You can associate the copy with a different model (you must give the copy a new component name). This allows you to use the existing attribute groups and attributes from the original component instance. You do not have to create them again or reassociate them.

Lock the component you want to copy, its descendants, and all instances of the parent of the newly created copy.

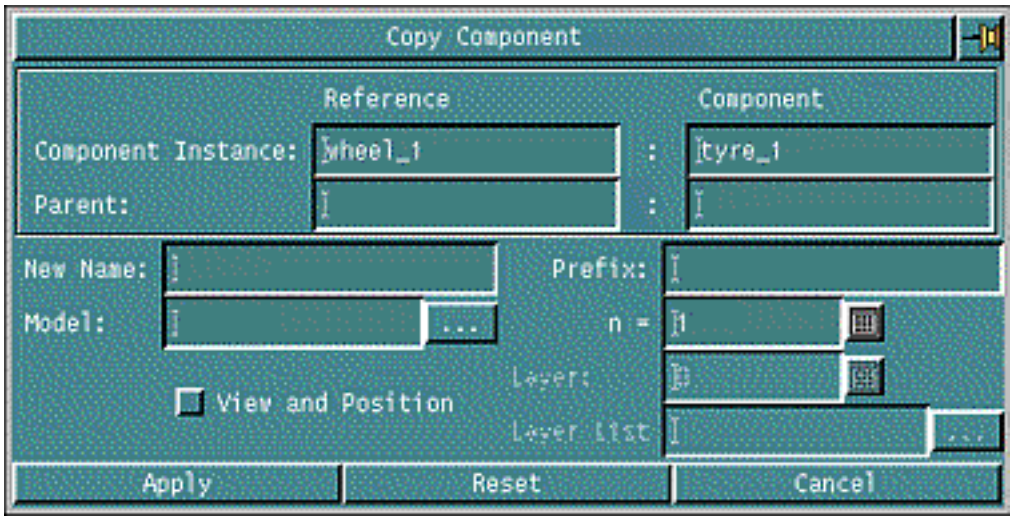
Please note: You can use the LOCK COMPONENT command to lock a desired component. (see “Determining What to Lock” on page 5-32).

Please note: If you have issued Select Assemble On (see “Automatic Assembly” on page 6-32), all parametric positioning constraints are solved when you copy a component.

To copy a component, do the following:

1. Choose the COPY COMPONENT option from the COMP/ASSY task set.

   Or, select the component instance you want to copy from the assembly structure and choose COPY COMPONENT from the pop-up menu enabled by the right mouse button. The COPY COMPONENT property sheet appears.



If you are copying a component in a reference assembly, specify input in the Reference column.

If you are copying a component in the main assembly, specify input in the Component column.

2. Enter the name of the component instance you want to copy in the Component Instance field in one of the following ways:
   - Select the component from the structure or select the model instance from the graphics window.
   - Type in the component instance name.
3. Enter the parent component to which you want the copies to be attached in the Parent field in one of the following ways:
   - Select the component from the structure using middle mouse button.
   - Type the name of the parent component in the parent field.
4. To give the copy a class name different from the original component instance, enter the new class name in the New Name field.
5. Specify the Prefix to be used while copying the component across assemblies to resolve naming conflicts.

6. To make multiple copies of the component instance, select the n= field. Use the calculator to enter the number of instances of the copy you want to make.
7. To associate the newly created component instance(s) with a different model, you must first change the component class name in Step 4.
8. To supply the (different) model name, type it in the Model field. Or, to generate a list of models in your CVPATH, click the List button, and then select the model name from the list that appears.

Please note: The View and Position options enables you to see and locate the associated models. See "Viewing and Positioning the Component" on page 3-12 for more information.

Positioning the Copied Component

When you copy a parent component and its children, the parametric positioning constraints between its child components are also copied. However, the constraints defined between the subassembly (or component) being copied and other subassemblies (or components) are not copied.

For parametric positioning, you must specify a location to initially view the copied component. You can then specify multiple parametric positioning options. For example, you can specify the Mate option once, followed by the Align option twice, before executing the command. The applicable positional constraints are solved and the component is copied parametrically.

Your input methods (syntactically) are:

```
COPY COMPONENT Parametric <loc> Mate Offset <real> Plane <face>
<face>
COPY COMPONENT Parametric <loc> Mate RevSurface <face> <face>
COPY COMPONENT Parametric <loc> Align Offset <real> Plane <face>
<face>
COPY COMPONENT Parametric <loc> Align RevSurface <face> <face>
COPY COMPONENT Parametric <loc> Align Edge <edge> Flip <edge> Flip
COPY COMPONENT Parametric <loc> Align Point <point> <point>
COPY COMPONENT Parametric <loc> Orient Angle <angle> Plane <face>
<face>
COPY COMPONENT Parametric <loc> Insert <face> Hole <face>
COPY COMPONENT Parametric <loc> Fix Ciname <node> <cplane> Ciname
<node> <cplane>
```

Mirroring a Component

You can mirror a component or a subassembly about a plane using the MIRROR COMPONENT command. You can perform the mirroring in two different ways:

- Part mirroring: Here the left-handed parts are created and associated with mirrored component.

  You can create a mirror copy of the source component. The left-handed (mirrored) parts associated with the created mirror copy of the component do have associative relationship with the parts of original component. You can disassociate these mirrored parts optionally.(For details, please see Update Ppart and Disassociate Ppart sections from *Parametric Modeling User Guide and Menu Reference*) During mirroring, both the structure as well as the graphics are mirrored.

  When you mirror a subassembly containing position constraints, these constraints get transferred to the mirrored subassembly.

Please note: The component instance to be mirrored should not contain any reference instance. In addition if the assembly to be mirrored has parametric positioning constraints, the mirrored parts should be created using either INSERT PPART command with mirror option in Parametric Modeling environment or this MIRROR COMPONENT command in Parametric Assembly environment, otherwise the positioning constraints do not get transferred to the mirrored part, though the mirroring does take place. This happens if you use legacy mirrored parts for the Mirroring command.

- Graphical mirroring: Here the orientations of the mirrored component are changed so that it appears mirrored, though the associated part is same as that of source component.

Please note:

1. In Parametric environment, both part mirroring and graphical mirroring is available, whereas in Explicit environment only graphical mirroring is available.
2. Orientation discrepancies may occur when you mirror the viewed components if they contain quarter circular or rectangular cut-off duct lines with quarter circular or rectangular cut-off fittings.

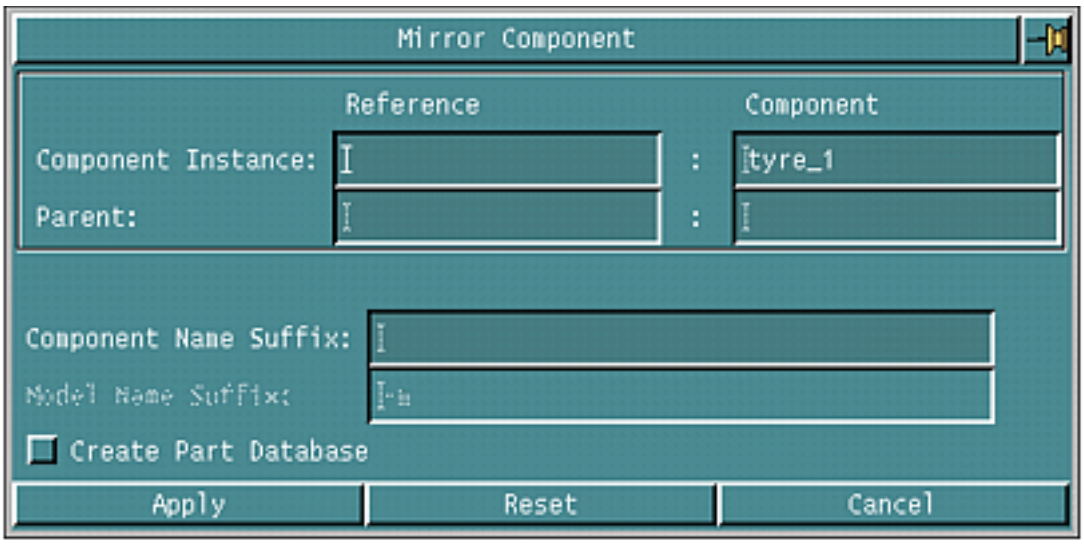
The mirrored copy of the component includes all the characteristics of the original component including associated viewing attributes and values. For example, the mirrored component uses the same layer for viewing as that of original component as an inherited attribute, unless a different layer is specified while mirroring.

To mirror a component, do the following:

1. Choose the MIRROR COMPONENT option from the COMP/ASSY task set.

The Mirror Component property sheet appears.

Figure 7-2 Mirror Component property sheet



If you are mirroring a component into a reference assembly, specify input in the Reference column.

If you are mirroring a component in the main assembly, specify input in the Component column.

2. Enter the name of the component instance you want to mirror in the Component Instance field in one of the following ways:
   - Select the component from the structure or select the model instance from the graphics window.
   - Type in the component instance name.
3. Enter the parent component name to which you want the mirrored component to be attached in the Parent field in one of the following ways:
   - Select the parent component from the assembly structure using the middle mouse button.
   - Type the name of the parent component in the parent field.

   If you do not specify any parent component, the mirrored component is attached to the parent of the original component.

Please note: The assembly to which the parent component instance belongs, has to be active.

4. Specify the suffix to be appended to the original component class name, at the Component Name Suffix field. If no suffix is specified, the original component instance name is used as the class name of the mirrored component.

Please note: If the generated mirror component name exists in the active assembly, the MIRROR COMPONENT command is terminated.

5. Specify the suffix to be appended to the original model name, at the Model Name Suffix field. If no suffix is specified, '-m' is appended to the original name to generate mirror model name.

Please note: If the generated model name exists in the directory(ies) specified under the active CVPATH, the MIRROR COMPONENT command does not perform geometric mirroring but only associates the new model name. If Create Part Database option is not selected, this input is disabled.

You must lock the component to be mirrored, along with its descendants, and the parent instance of the newly created mirrored copy. Lock the parent component instance of the original component.

Please note: You can use the LOCK COMPONENT command to lock the desired component.

6. Click on Create Part Database option if you want to perform part mirroring. (Default is set to OFF, allowing only graphical mirroring).

Please note: In the parametric environment the Create Part Database option can be set ON or OFF, but in explicit environment it is always hazed out.

7. Click Apply. The following menu appears.

Figure 7-3 Mirror Component menu

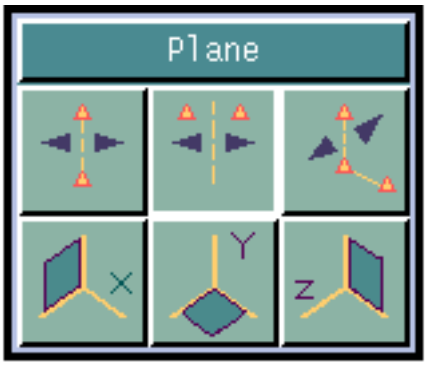
You can cancel the mirroring by selecting Abort option from this menu.

8. Click the Plane option to get the plane selection menu.

   Here different menus appear for Parametric and Explicit Assembly environments.

- Parametric plane selection:

Figure 7-4 Parametric Plane Selection menu

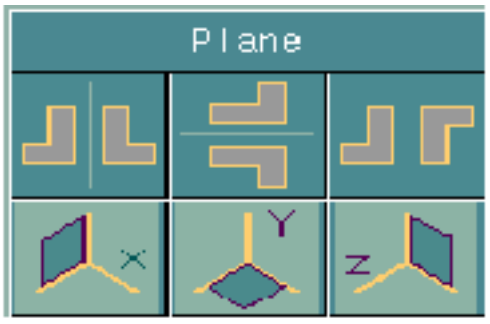


Refer to the *Parametric Modeling User Guide and Menu Reference* for details on specifying planes.

To perform mirroring about the x-, y-, and z- planes, (these are the active Cplanes), you need to specify a location through which the plane passes.

- Explicit plane selection:

Figure 7-5 Explicit Plane selection menu



The first row icons are similar in their behavior to the MX, MY and MB orientation options respectively, of Explicit modeling environment.

With the MX and MY options, you can perform mirroring about the x- and y-planes of the source components Cplane. With MB option, first the MX mirroring is performed and then, the MY mirroring is done.

You do not need to specify the location. The location is the origin of the source component or subassembly.

The second row icons are similar in their behavior as explained in parametric plane selection above.

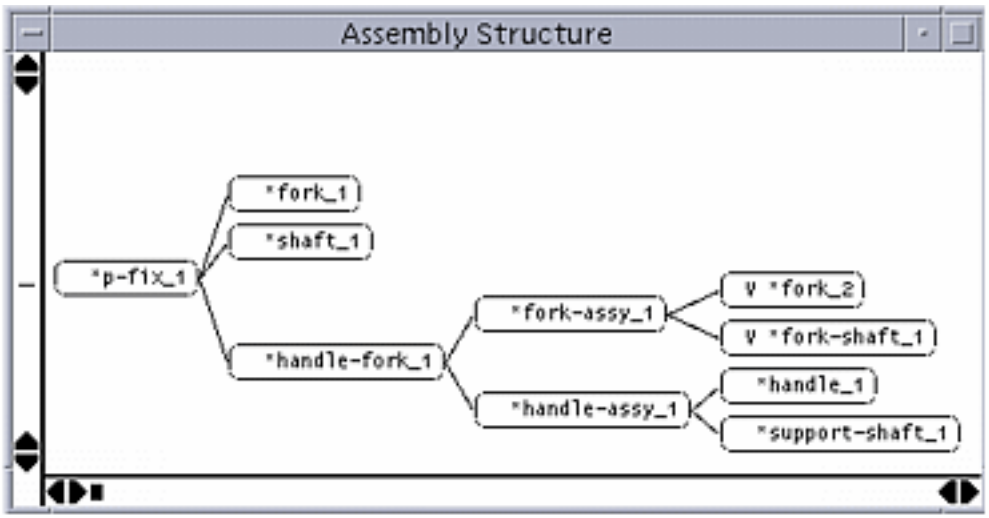
9. Select Done to complete the mirroring of component.

Please note: For part mirroring, you can use old CADDS parts for mirroring but the names of the mirrored part should follow the '<source partname><suffix>' format. (You can do this by renaming the old CADDS parts to the desired format.) Here the system searches for the specified part name in the directories specified under the active CVPATH. You may not get expected result with the old mirrored parts.

Example: Mirroring a Component in the Parametric Environment by Creating a Part Database

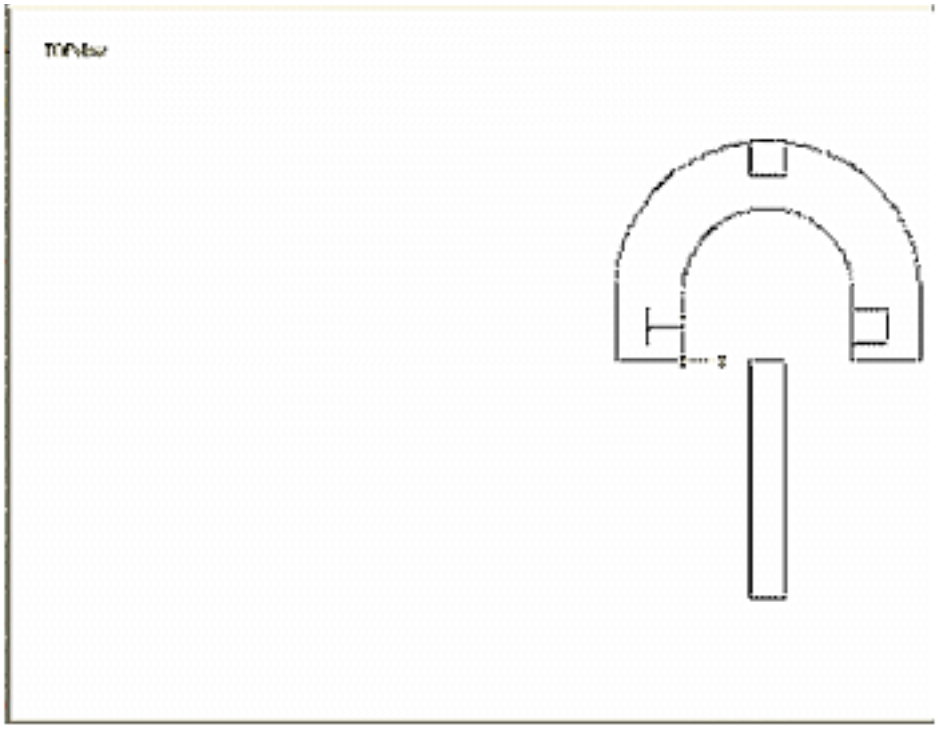
Shown below is a assembly structure of component `fork-assy_1` containing instances of `fork_2` and `fork-shaft_1`. Suppose you want to mirror copy `fork-ass_1` with `handle-fork_1` as parent component, by specifying a mirroring plane using 2 points.

Figure 7-6 Assembly Structure before mirror copying



The top view of the actual geometry related to component `fork-assy_1` is shown below.

Figure 7-7 Geometrical representation of component fork-assy_1



Given below is the procedure to mirror and copy the component `fork-assy_1`:

1. Choose the MIRROR COMPONENT option from the COMP/ASSY task set.

   The Mirror Component property sheet appears, see Figure 7-2 on page 7-16.

2. Select the instance `fork-assy_1` from the assembly structure widow, using left mouse button. This specifies the component to be mirror copied.

   OR

   Enter the Component name for Component Instance as `fork-assy_1`.

3. Select the instance `handle-fork_1` from the assembly structure window, using middle mouse button. This specifies the parent component to which the mirror copied component is to be attached.

   OR

   Enter the Component name for Parent as `handle-fork_1`.

4. Enter the Component Name Suffix as `-lhs`.

5. Select the Create Part Database option.

6. Enter the Model Name Suffix as `-lhs`.

7. Choose Apply.

   The Mirror Component menu appears, see Figure 7-3 on page 7-17.

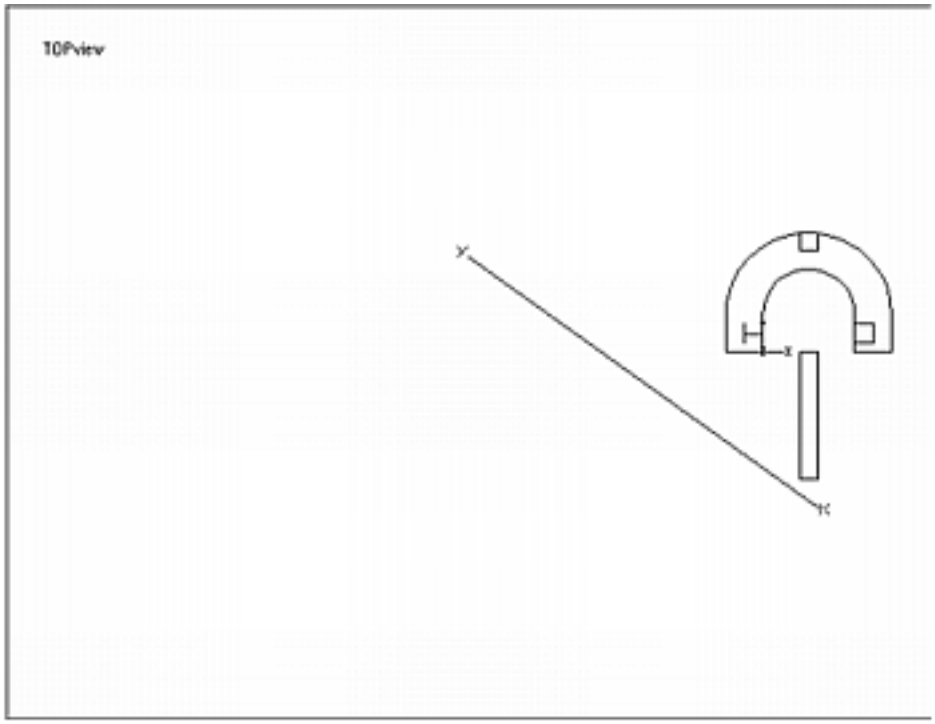
8. Choose Plane.

   The Plane menu appears, see Figure 7-4 on page 7-18.

   a. Choose the first option from Plane menu to specify the plane perpendicular to the active Cplane and passing through the two specified locations.

   b. Select two points, as shown below.

Figure 7-8 Two points specifying the mirror plane

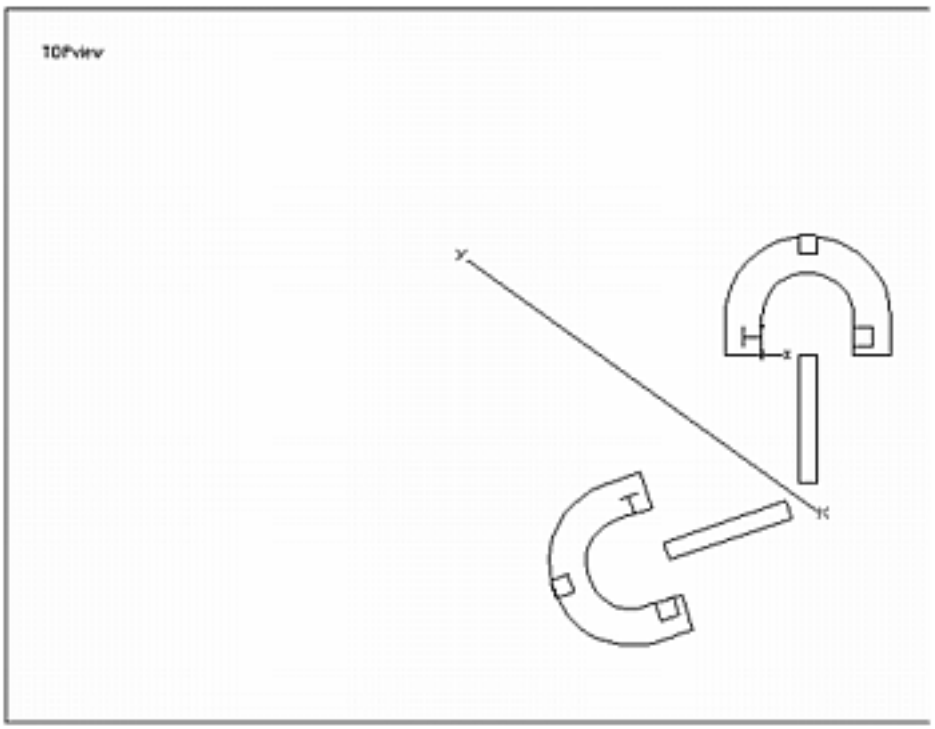


A line appears, which denotes the plane normal to xy plane.

9. Choose Done.

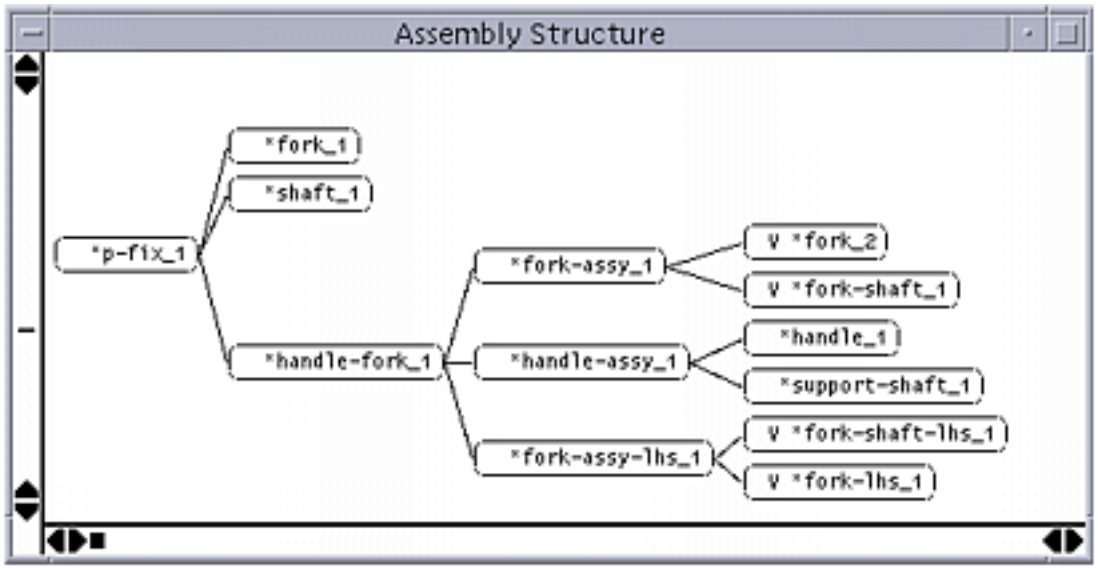
The mirrored copy of the original component appears as shown below.

Figure 7-9 Final mirror copied geometry



The original assembly changes to reflect the mirroring and copying of the component `fork-assy_1`. This mirror copied component is attached to the parent component `handle-fork_1` with suffix `-lhs`, as `fork-assy-lhs_1`. The child components `fork_2` and `fork-shaft_1` of the original component are also mirror copied with the suffix `-lhs` as `fork-shaft-lhs_1` and `fork-lhs_1`.

Figure 7-10 Final Assembly structure after mirror copying



To execute the above steps, you can give following command on the command line:

```
Mirror Component Ciname fork-assy_1 PARENTCiname handle-fork_1
Newcnamesuffix -lhs Modelnamesuffix -lhs Copy Yes Plane Axis Free
loc[-10,0,0] loc[3,-10,0] Go
```

Specifying Orientation for Mirror Copied Component

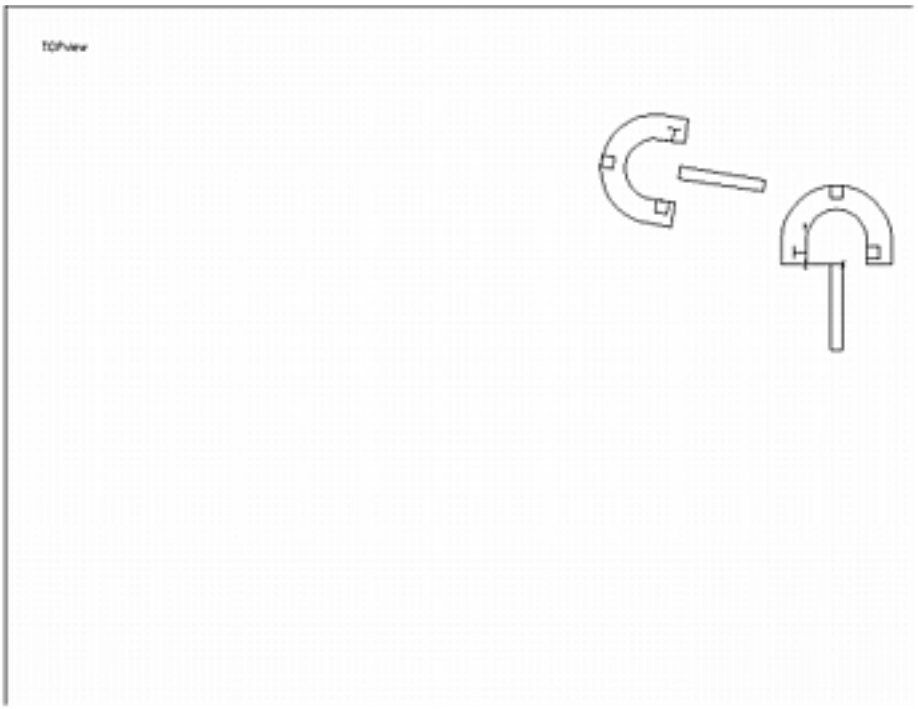
Once the component is mirrored you can use the Move Component command to position and orient the mirrored component. You can also orient the mirror copied component by executing the Mirror Component command from the command line and specifying the position and the modeleroption details. For details refer the online command help for the command MIRROR COMPONENT.

With the same example, suppose the mirror copy is to be positioned at location [-10, 10, 0] and rotated about Z axis through the location [-10, 10, 0] by an angle of -45.00 degrees. To do so, give the following command at the command line:

```
Mirror Component Ciname fork-assy_1 PARENTCiname handle-fork_1
Newcnamesuffix -lhs Modelnamesuffix -lhs Copy Yes Plane Axis Free
loc[-10,0,0] loc[3,-10,0] Modeleroptions yes loc[-10,10,0] Rotate
Angle -45.0000 Axis Z loc [-10,10,0] Go
```

The resultant geometry after the mirror copy has been viewed, is shown in the figure below.

Figure 7-11 Mirror copied geometry with specified orientation



Changing a Reference Assembly

Choose the Change Assembly option from the COMP/ASSY task set to make changes to a reference assembly that appears within your active assembly. With the Change Assembly option you can do the following:.

- Detach the reference assembly from its old parent, and attach it to a new parent.
- Change the name of the reference assembly as it appears in your active assembly.(The name of the external assembly does not change.)
- Change the instance name of the reference assembly as it appears in your active assembly.
- Activate and change any aspect of the reference assembly within an active main assembly.
- Change the revision of the reference assembly.
- Completely change the external assembly.

Please note: You can activate and edit a reference assembly from the active main assembly by locking a child component of the associated assembly instance. The act of locking any child of a reference assembly instance activates the associated assembly database (`_db`) (see “Editing an External Assembly” on page 7-27).

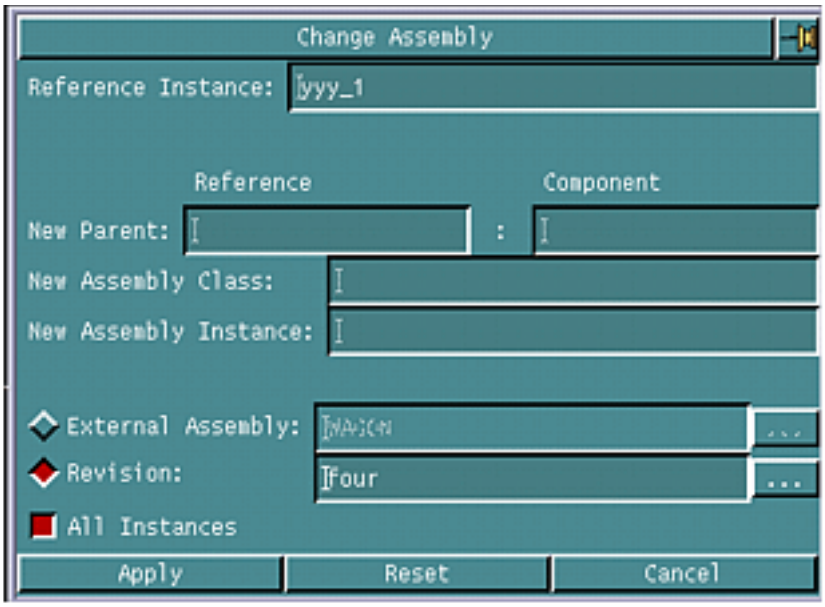
The root node of a reference assembly is called an assembly instance. It is seen in a bold line font. It is an assembly instance, not a component instance (see “Adding Existing Assemblies to Your Assembly” on page 3-21).

You can change the structure, orientations, attributes, and model geometry associated with components in a reference assembly, provided you can lock the impacted components or attributes.

Please note: The CHANGE ASSEMBLY command is valid for reference assembly instances, and not to the assemblies added with the Copy option of Add Assembly. When you add an assembly as a copy, it becomes a subassembly of the active assembly, all though you had created it using Add Component. There is no association back to the original assembly on disk. Use Change Component option to make changes to a subassembly added with the Copy option.

The Change Assembly property sheet appears with following options:

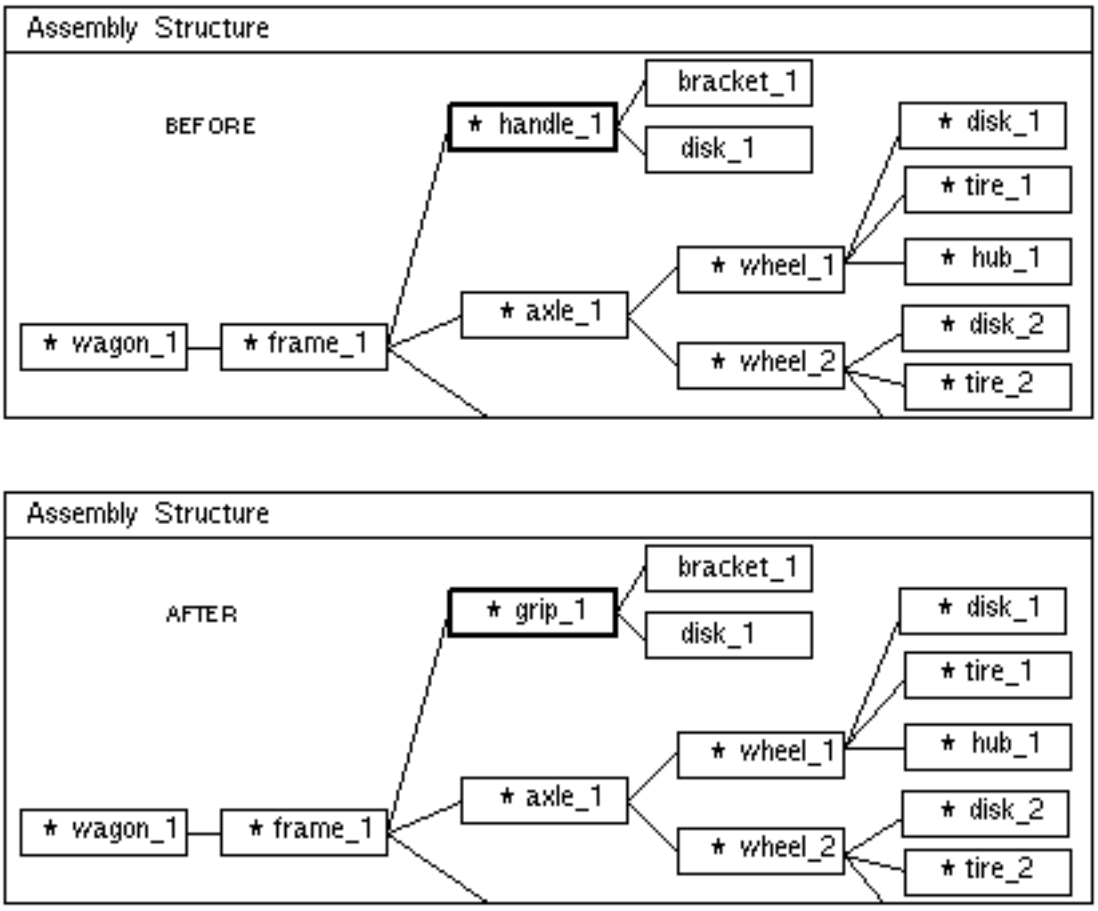
Figure 7-12 Change Assembly property sheet



- Specify the name of active reference assembly in the Reference Instance field.
- Specify the component name or reference assembly, which is to be the new parent of the active reference assembly, into the New Parent field.
- Specify the new class name in the New Assembly Class field in case you want to change the class name of the selected assembly.
- Specify the new instance in the New Assembly Instance field in case you want to change the instance of only the selected assembly.
- You can specify a different external assembly name than the one with which the current active reference assembly is associated, in the External Assembly fields. You can select the one from the available runtime list. (Here you can choose the active CVpath from the run time list.)
- Specify the revision of the associated external assembly in the Revision field. You can choose the revision name from the available runtime list. You can choose either the external assembly or the revision at a time.
- The All Instances option is automatically selected if choose to change the external assembly or the revision, but if only the parent is to be changed it is set to a particular instance and not to all instances.

If you are changing an assembly in a reference assembly, specify the assembly instance name in the Reference field.

In the example, the user has changed the name of the reference assembly from `handle` to `grip`.



Please note: You do not need to activate the referenced assembly to perform this change.

Editing an External Assembly

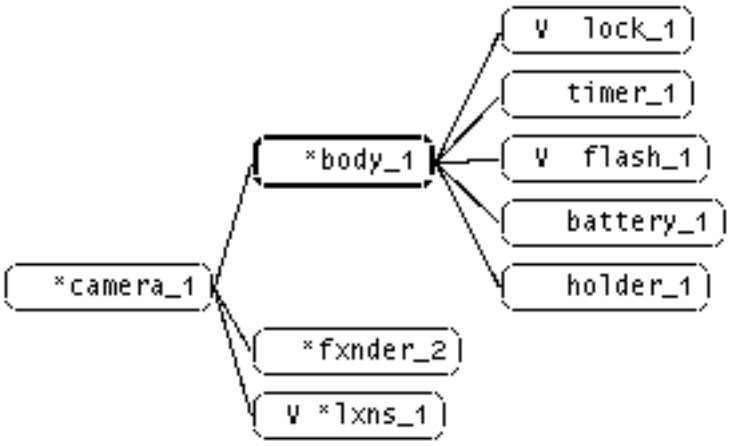
To perform modification in a reference assembly, the reference assembly must be activated. The first time you lock a component in a reference assembly, the assembly instance (root component of referenced assembly) is automatically locked and activated. Once the assembly is active, it becomes a concurrent assembly. Everyone else active in the same assembly can see other concurrent changes through the Refresh Assembly capability. Everyone can establish locks on components to prevent modifications by others.

By default, three reference assemblies can be active at the same time in an assembly session. Use the CV_MAX_ACTIVE_REF variable to set the maximum number of active reference assemblies in an assembly session.

When the limit of active reference assemblies is reached, you can use the command QUIT REFERENCE to deactivate active reference assemblies (see “Deactivating a Reference Assembly” on page 4-3).

In this example, the user changes the external assembly association of the `body_1` assembly instance to revision `2` of the same assembly name (`body-2/_db`). All instances of `body` are affected.

Please note: The assembly instance `body_1` must be locked.



Understanding System Messages

The first time that you lock a component in a referenced assembly (LOCK COMPONENT) the system activates the associated external assembly. At this time the following message appears:

```
Activating reference assembly %assembly name%, please wait!
Activation process is complete!
```

Before the tree activation process occurs, if the current version of the reference assembly tree is not up-to-date, the command is aborted with the following error message:

```
WARNING: Another user has made changes to the active assembly
tree. In order to perform this command, you must issue a REFRESH
ASSEMBLY command to get an updated version of the assembly tree.
```

Changing Structure Appearance

When your assembly structure is large and complex, you can simplify its appearance by compressing or blanking the display of subassemblies. Compress a subassembly to show multiple instances of leaf components as a single component. Blank a subassembly to show only its parent component.

Use the DISPLAY STRUCTURE option from the popup menu, or select the icon from the COMP/ASSY task set.

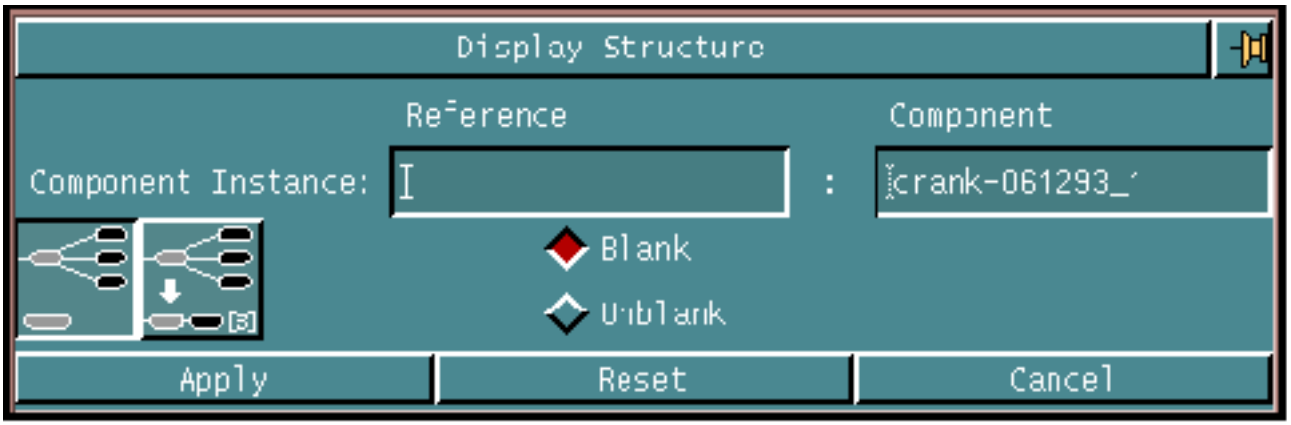
Please note: If an assembly is activated in the Assembly Tree Window, then:

- The Display Structure menu is unavailable.
- The information about hidden sub assembly structures (the result of DISPLAY STRUCTURE BLANKFLAG YES command) is not considered and the entire assembly tree structure is available for display.

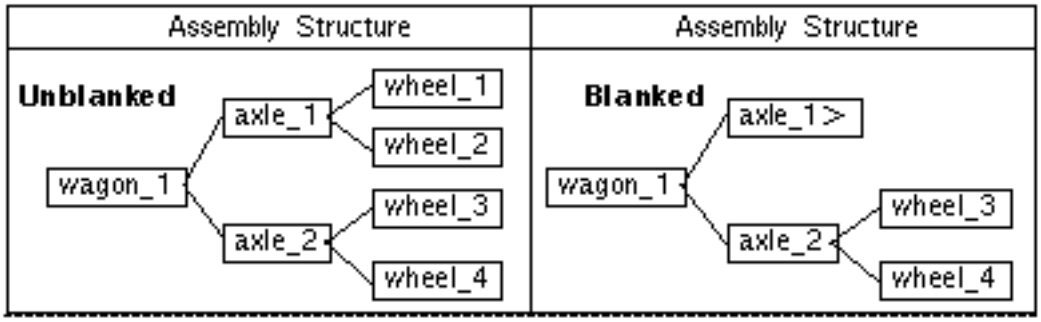
If you lock the parent of the subassembly before changing the display, the change is saved in the database. If you do not lock the parent, the change is for only the current session.

Blanking a Subassembly

Select the parent component of a subassembly or reference assembly to blank and click Apply. The structure displays only the parent component you specify. A greater than sign(>) indicates the component is blanked, and that components exist below it. When you unblank, all components below the parent reappear.



If you are specifying a Component in a reference assembly, specify the assembly instance name in the Reference field.



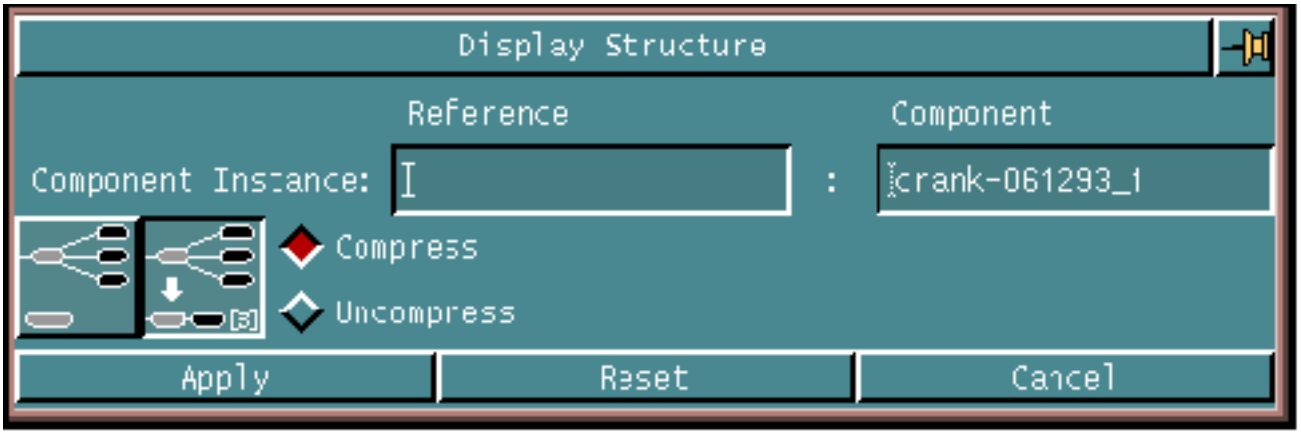
Compressing a Subassembly

When you compress a component instance or reference assembly instance, all leaf nodes that exist below the component and have multiple instances are compressed. A single component with a number represents the compressed leaf nodes.

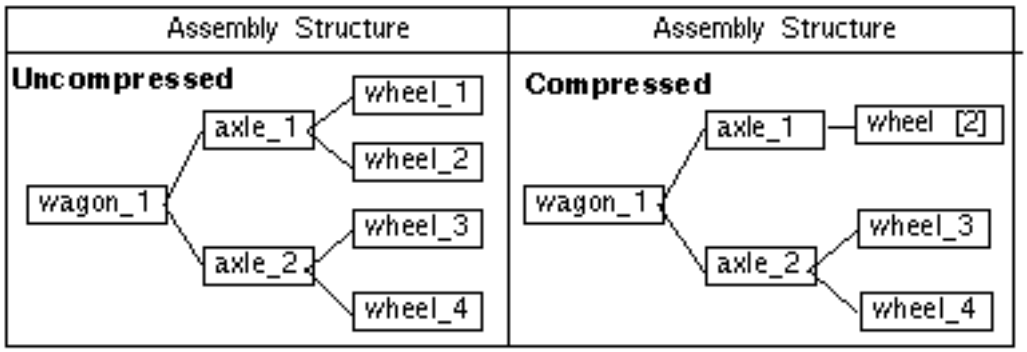
When you select Uncompress, all compressed nodes below the specified component instance are displayed with one component for each instance.

Fill in the property sheet by selecting a component instance or assembly instance from the Assembly Structure window, or by typing the name on the property sheet.

Please note: Component instance numbers must be in numerical order to be compressed (see “Reordering a Component” on page 7-32).



If you are specifying a Component in a reference assembly, specify the assembly instance name in the Reference field.



Reordering a Component

You can change the order in which sibling components appear in the assembly structure. For example, you can reposition components in the structure such that their instance numbering is consecutive or they are closer to mating components or subassemblies.

You can reorder components in the active main assembly, as well as components in an active reference assembly. Reordering does not change the position of the associated model(s). It only affects the appearance of the assembly structure.

The components you are reordering need not be instances of the same class. If the component is also the parent component of a subassembly, the entire subassembly remains attached to its parent (with the parent in its new position in the structure).

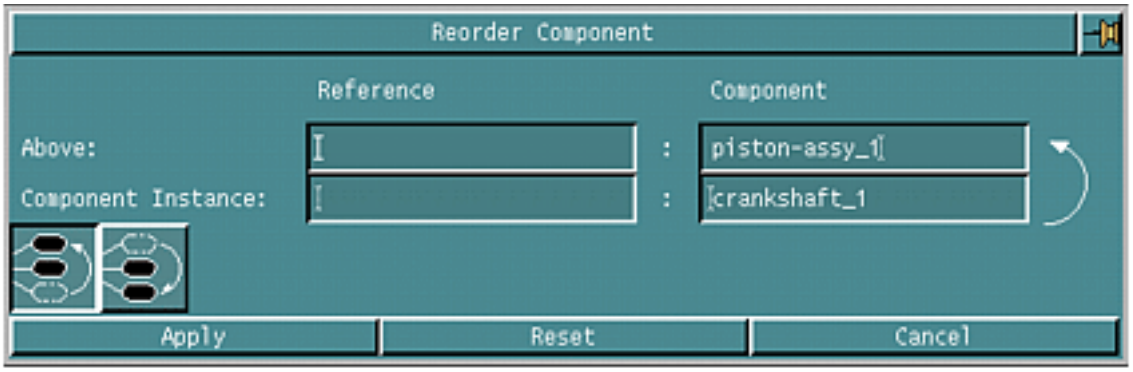
All affected components must be locked.

Placing a Component above Another in the Main Assembly

To place a component above a sibling component in the main assembly, follow these steps:

1. Open the REORDER COMPONENT property sheet.
2. Select the Above icon on the property sheet.
3. Using the left mouse button, select the component instance you want to reposition. Its name appears in the Component field in the Component Instance row.
4. Using the middle mouse button, select the component instance you want to place it above. Its name appears in the Component field in the Above row.

5. Click Apply. The component specified in step 3 is placed above the component specified in step 4.



If you are specifying a Component in a reference assembly, specify the assembly instance name in the Reference field.

Placing a Component above Another in a Reference Assembly

To place a component above a sibling component in a reference assembly, do the following.

1. Activate the reference assembly in the main assembly, by locking any one of its children components.
2. Open the REORDER COMPONENT property sheet.
3. Select the Above icon on the property sheet.
4. Using the left mouse button, select the component instance you want to reposition. Its name appears in the Component field in the Component Instance row. Its parent assembly instance name appears in the Reference field in the Component Instance row.
5. Using the middle mouse button, select the component instance you want to place it above. Its name appears in the Component field in the Above row. Its parent assembly instance name appears in the Reference field in the Above row
6. Click Apply. The component specified in step 4 is placed above the component specified in step 5.

Placing a Component below Another Component

To place a component below a sibling component in the structure, access the REORDER COMPONENT property sheet. Select the Below icon. Fill in the fields as appropriate. For reference see “Placing a Component above Another in the Main Assembly” on page 7-32 or “Placing a Component above Another in a Reference Assembly” on page 7-33.

Obtaining Information about Components

You can obtain information about a component in two key ways:

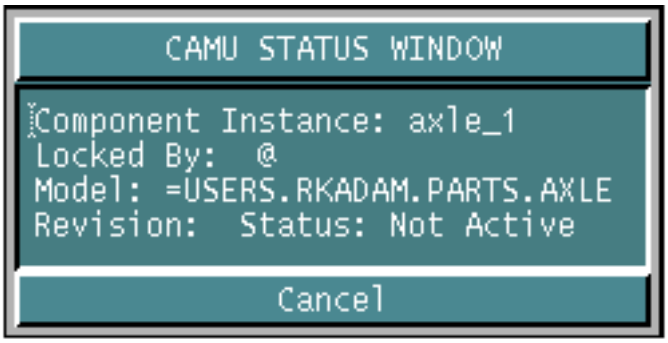
- Using the CAMU Status Window
- Using the Verify Object Function

Using the CAMU Status Window

The CAMU Status window appears above the Assembly Structure window. When you select a component in the assembly structure, the Status window is updated to display the following information:

- Selected component instance name
- Name of the user who has locked the component
- Associated model name and revision
- Associated model status (active, suspended, or inactive)

The CAMU Status window is as shown:



You can close the status window by clicking the Cancel option on the window. To launch the CAMU Status window, select the CAMU Status Window option from the Information Access icon (i) at the upper left of the screen.

On minimizing the CADDS window, the CAMU Status window also closes. You can view it again on maximizing the CADDS window. However, if you close the status window by choosing the Cancel option on the CAMU Status window, then it is not displayed again.

Please note: The CAMU status window is displayed along with the Assembly Structure Window only. That is, when the environmental variable `AD_C_SERVER_ENABLED` is set to `no`.

Using the Verify Object Function

VERIFY OBJECT provides a brief or detailed description of the component instance or assembly instance you select. The description appears in the text window or the Report Window.

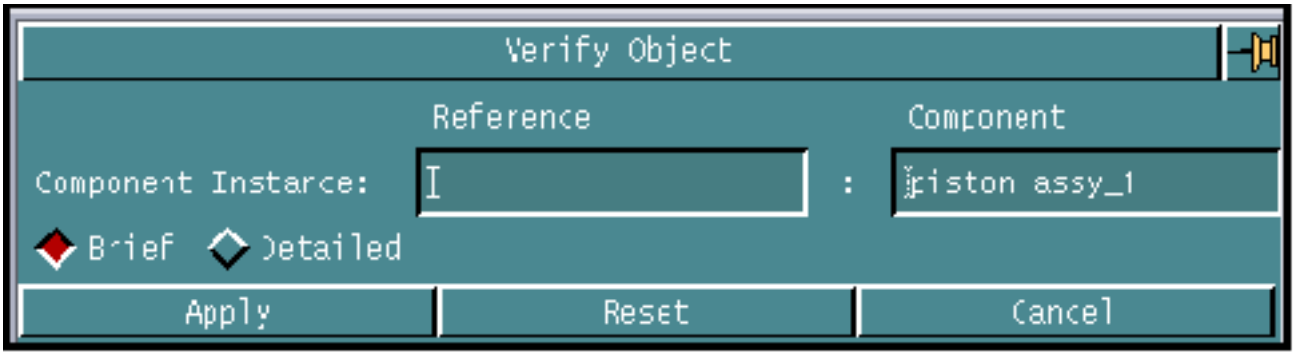
In the Parametric environment, the description appears automatically in the Report Window but does not appear in the text window. In the Explicit environment, the description of the component or components is by default displayed only in the text window. To view the description of the component or components in the Report Window in Explicit environment, set the environment variable in the `.caddsrc-local` file as follows:

```
setenv USE_EXP_REP_WINDOW 'yes'
```

The default setting for this environment variable is `no`.

The description includes the mirror status of the component, along with the plane about which the mirroring is performed. In the Parametric environment, you get details of part mirroring, a list of inserted source part names and out of sync source part information.

Please note: The details relating to inserted source parts and out of sync source part appear only if the model associated with the component being verified, is created using the Insert Ppart command. Refer *Parametric Modeling User Guide and Menu Reference* for details on resolving out of sync source parts.



Reference: Specifies a reference assembly (assembly instance) to verify. When used with Component, specifies the referenced assembly containing the object.

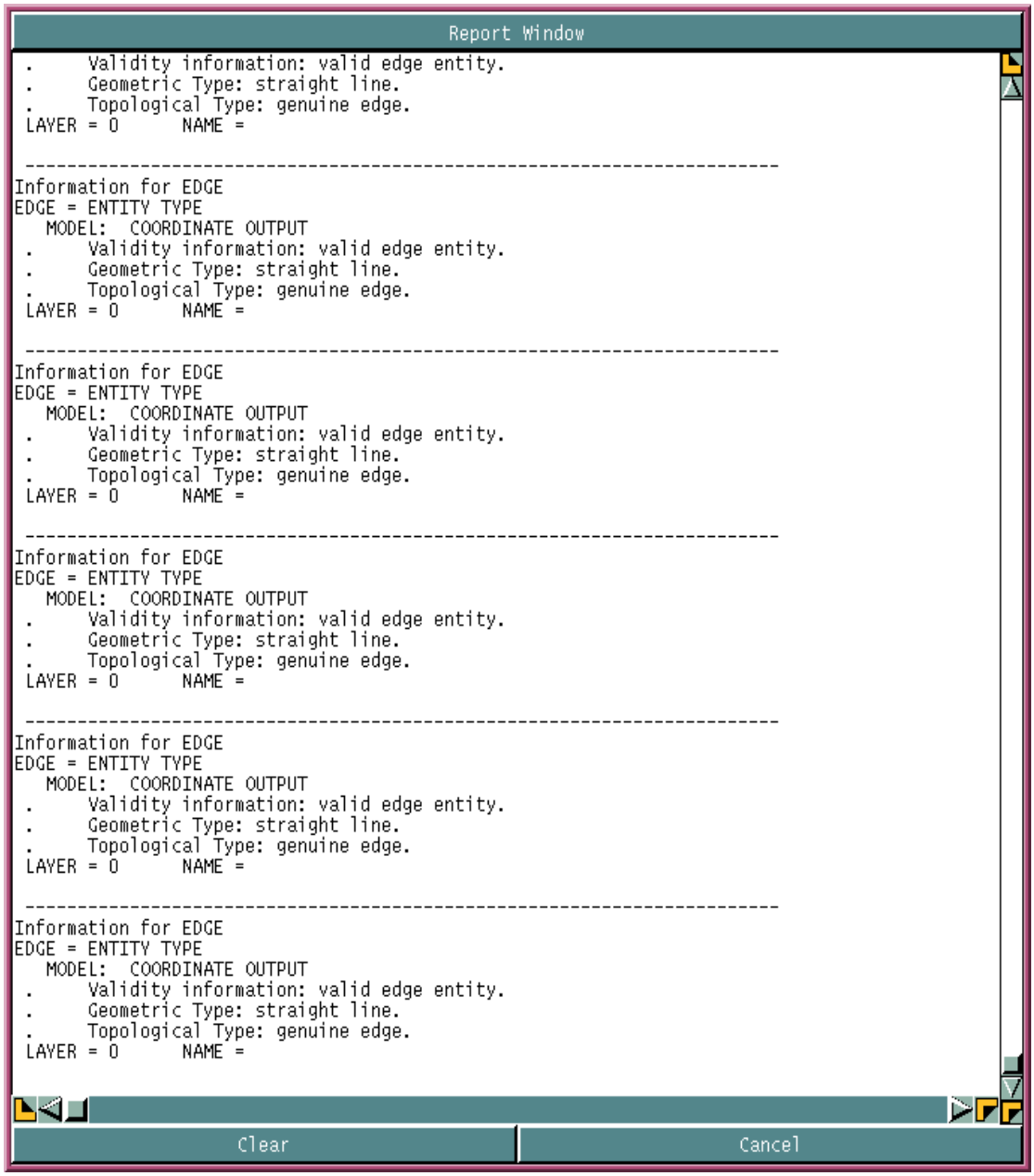
Component: Specifies the component instance name to query.

Brief : Displays the assembly name, object type, ID number, lock status, associated model name, and whether the component is viewed.

Detailed : Additionally displays the class name and the ID number, child relation names, and information about attribute groups associated with the component instance or reference assembly instance.

Please note: If the model associated with the instance is viewed, the Detailed option also provides location and orientation information.

The Report Window appears as follows when accessed from the UTILITY pull down menu:



Sample output of the Report Window is shown below:

```
Assembly name: =USERS.SPATIL.PARTS.CAMERA
Object type: component instance
ID number = 116
Associated Model: =USERS.SPATIL.PARTS.CAMERA.FRAME
Lock Status: not locked by anyone
Model Status: active by prevelas
Layer Offset = 0
Layer List = 0-254
IDname: 116
References: FRAME
Origin: OX = 0.000000 OY = 0.000000 OZ = 0.000000
Orientation: AX = 0.000000 AY = 0.000000 AZ = 0.000000
Graphically Mirrored: No
Blanked: No
Hilited: Yes
Part Mirrored: Yes
Source Part Name: disk
Part Out of Sync: Yes
```

Listing Reference Assemblies

LIST ASSEMBLY lists all active and nonactive reference assemblies sourced in the active main assembly.

The command gives the following information:

- Active main assembly name
- All reference assembly names
- Active or non active flag for all reference assemblies

Chapter 8

Defining Assembly Geometry

This chapter describes how to view and position the models associated with assembly structure components in the context of the assembly. The topics are:

- Guidelines for Creating Model Geometry
- Working on Model Geometry
- Component Instances and Their Locations
- Locating Model Space Origin for a New Model
- Viewing a Component's Associated Model Geometry
- Moving a Component
- Highlighting the Appearance of Geometry

Guidelines for Creating Model Geometry

- Use solid or surfaced double precision or C5 models. Surfaced models can be either ASD or Nurbs trimmed surfaces.
- All models in the assembly must be of the same precision.
- If 3D wireframe models exist, associate the relevant component in the assembly structure with them. Rebuild them as 3D solids or surfaces as time permits.
- Each model should contain a current explicit graphics file (`_gr`) containing the drawing and model graphics data along with the tessellation data. Ensure that the tessellation data stored in the `_gr` file is in sync with the model.
- Use trimmed rather than block surfaces, that is, oversize surfaces with the profile of the panel or piece projected onto it, to enable accurate clash detection and visualization.
- You do not need to create model geometry using Nurbs, but the process of trimming should yield Nurbs surfaces as output.
- All surface models should have a small mesh suitable to the shape of the surface (that is 2x2) before filing to prevent confusion when viewing large models and assemblies.
- Blank all model geometry not required for viewing by the rest of the team (that is, construction geometry) before filing the model.
- All components should have entities to assist with their location in an assembly. For example, lines down the centers of bores and shafts where they mate with other components and lines along the X,Y, and Z axes of the part datum (model space origin of the model).
- Create a mass point for the solid or surface in the final solid or surface. You can use these from within CAMU.

Working on Model Geometry

To work on the model geometry associated with a particular component class name, select the component and activate the model. Any changes you make to the model are reflected in all instances of that component class.

When you work in an assembly, you work in either an active Adrawing or an active model. In the Adrawing, you can change the position of component instances in the context of the entire assembly. In the Adrawing, however, you cannot work on the actual model geometry associated with different components in the assembly. You must activate the model you wish to work on. This enables you to create or edit its geometry.

A New Component and Its Associated Model

When you create a component in the Assembly Structure window, a model name identical to the component class is automatically associated with it. The associated model may or may not exist.

You can also associate the component to either a new or an existing model when you create a new component. You can optionally associate the model with a family member of a master model (see "Adding a New Component to the Structure" on page 3-9). You can change the model association with Change Component (see "Changing a Component" on page 7-4).

A newly created component is locked automatically. To activate and work on the model of an existing component, the component must be locked.

Please note: You can use the LOCK COMPONENT command to lock a desired component. (see "Determining What to Lock" on page 5-32).

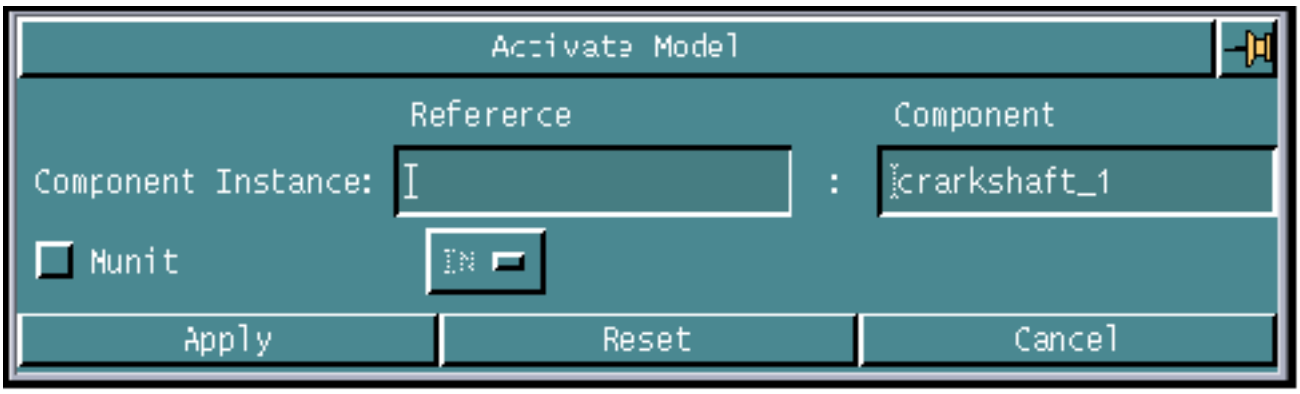
Activating a Model

To create or edit geometry in a model associated with a component class in the assembly, do the following:

1. Select the component from the assembly structure. This highlights the component.
2. Place the cursor inside the Assembly structure window and click the right mouse button to display the context-sensitive menu. Choose the Activate Model option or select the Activate Model icon from the resultant menu.

   The Activate Model property sheet displays the name of the highlighted component in the Component column and Component Instance row.

If the model exists, Munit is ignored. If Activate Model is creating a model, Munit determines the model units. Use the model unit defaults, or choose Munit and then select the model units you want.



Use the Reference option to specify a reference assembly (assembly instance). When used with Component, this specifies the referenced assembly containing the component whose associated model you wish to activate.

3. Apply the property sheet to activate the associated model.
4. Select the modeling environment (Explicit or Parametric) and task set in which to work.

When a model is activated, the view is inherited from the Adrawing you were in when you selected Activate Model.

5. Define or revise model geometry using the same procedures as when you activate a part in the CADDS Parametric or Explicit modeling environments. For more information, see the *Parametric Modeling User Guide and Menu Reference* and the *Explicit Modeling User Guide and Menu Reference*.

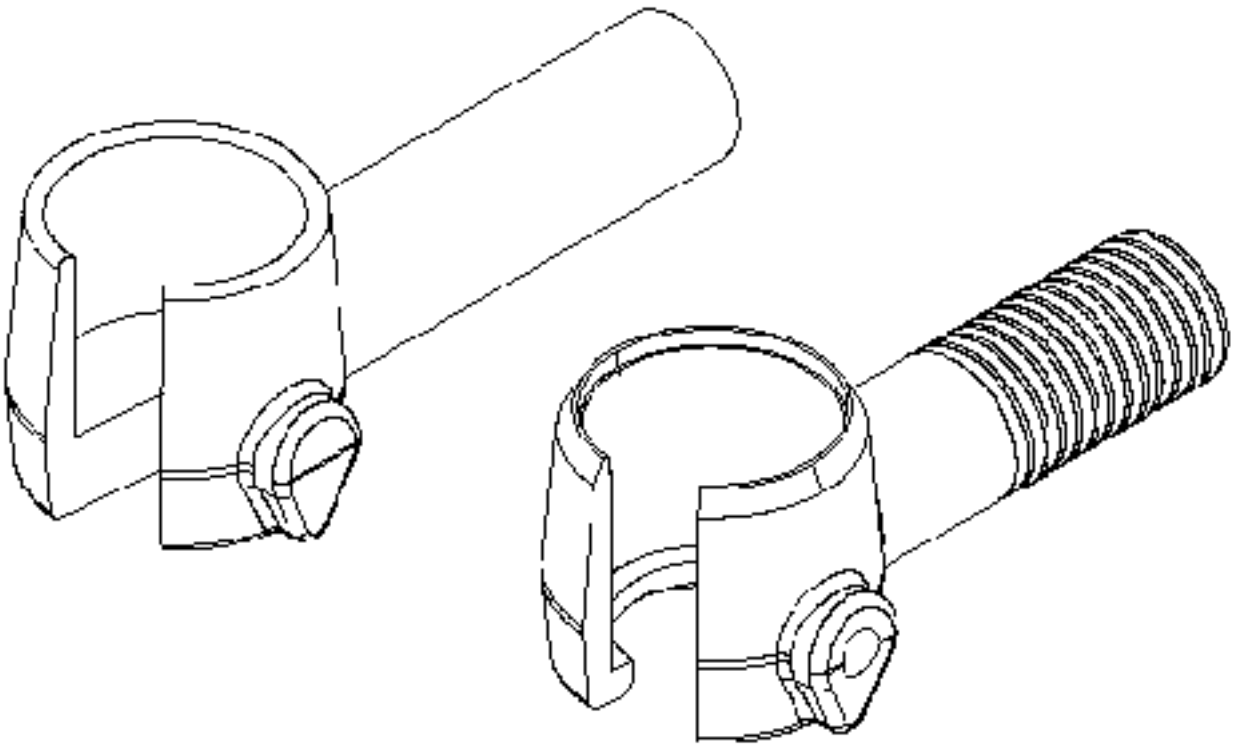
Using a Source Model to Create a New Model

You can use the Insert Ppart command to create a new parametric model (target) using some or all of the geometry in an existing parametric model (source). You can also use the Update Ppart and Dissociate Ppart commands.

Insert Ppart enables the following scenarios:

- Multiple designers can work independently on different sections of the same piece part. They can then merge these parts together into one parametric model.
- The same parametric part can be used to define or drive both a finished part and the manufacturing counterpart (for example, mold cavity) used to create it.

For example, a forging model is created. Additional operations must then be performed on the model to generate a machined part. Any modifications to the forging must be propagated to the machined part.



Associativity between the source and target models is automatic. Changes to the source can be seen in the target as follows:

1. Activate the source model.
2. Make modeling changes and file the model.
3. Activate the target model.
4. Issue the Update Ppart command.

For a complete description of this functionality, see the *Parametric Modeling User Guide and Menu Reference*.

Component Instances and Their Locations

When you create component instances, you have the option of viewing and positioning them in the active Adrawing.

If you do not position the instances when you create them, you can use the Move Component icon. Using Move Component option, you can Translate, Rotate, Orient, and Position.

You can view components before positioning them. They are viewed at their default locations, and you are prompted for additional information to locate them.

The ways you can change the location of component model instances depend on whether you are in the Parametric or Explicit environment. When you change the location of a component instance in the graphics display of the assembly, the component you want to position and its parent component must be locked.

Translate, Rotate, Orient, and Position Component are all available in both the Assembly in Explicit and Assembly in Parametric environments. None of these positioning tools can operate on the root component, the active model, or ancestors of the active model. Neither can they operate on component instances of a reference assembly, except in an Exploded View state.

Please note: Model instance positions are constant across all Adrawings in the assembly.

Locations of Components in Reference Assemblies

You cannot reposition a component instance in a reference assembly unless the Exploded View option is turned on. For more information, see Chapter 10, “Adrawings and Alternate Views”.

Locating Model Space Origin for a New Model

Specifying the model space origin (MSO) of new models before creating them simplifies the layout, design, and assembly process for the entire design team. It also eliminates problems for other members of the design team who may wish to use the same model in different assemblies.

If the MSO of a model is not appropriate for use in another assembly, you can activate the model and move its geometry onto MSO in the assembly. However, moving geometry at the piece part level shifts out of position any component instances in other assemblies that are also associated with this model. Whenever you translate geometry at the piece part level, it affects the position of all component instances associated with that model, in all assemblies containing that component instance.

Specify the MSO location for any model in your assembly before you create the model database; otherwise it is at the same location as the MSO of assembly itself.

Creating A New Model without Locating Model Space Origin

The example that follows (pulley assembly) shows how to create and locate a new model in an assembly without first specifying the MSO location of the new model.

Pulley Assembly Structure

Assume the `DEFAULT` Adrawing is active and the assembly structure has already been created. At this point, none of the required models exist. The leaf nodes in this assembly structure are associated with nonexistent models.

Figure 8-1 Pulley Assembly Structure

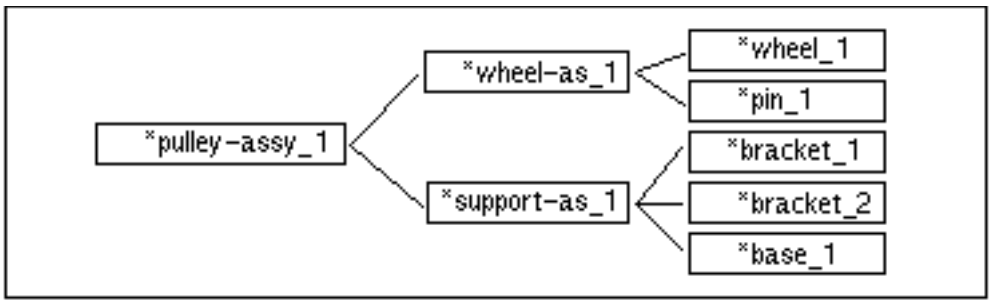
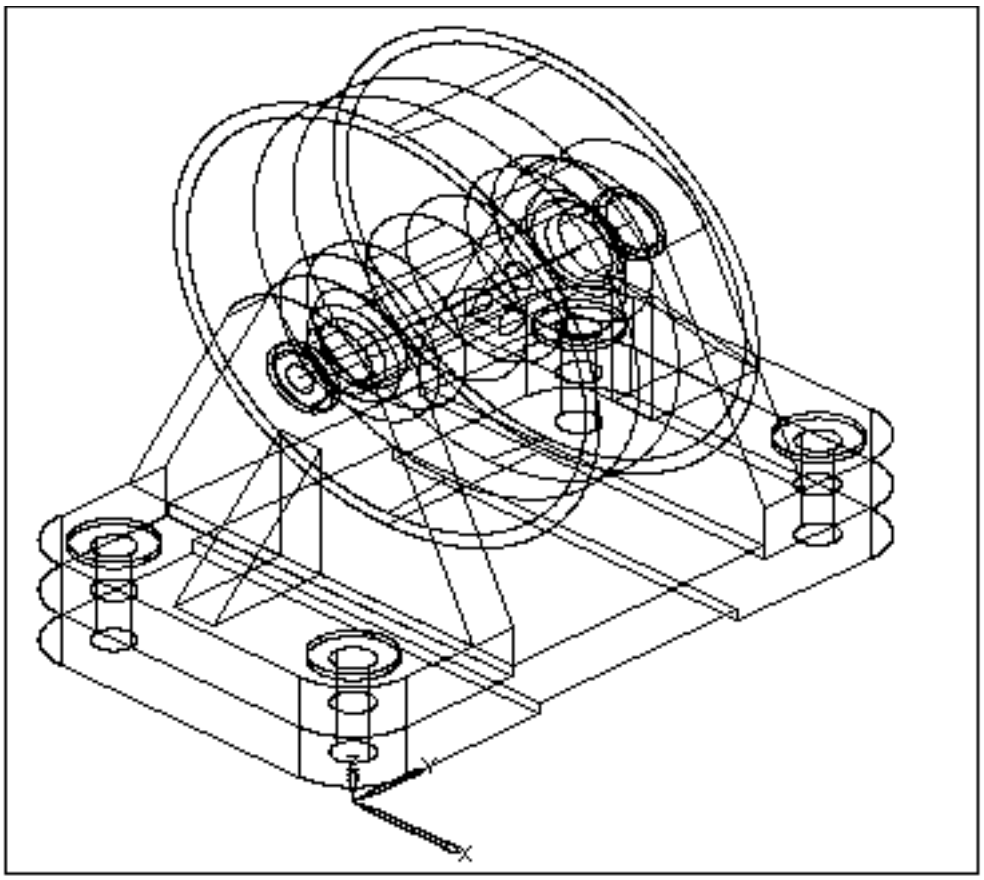


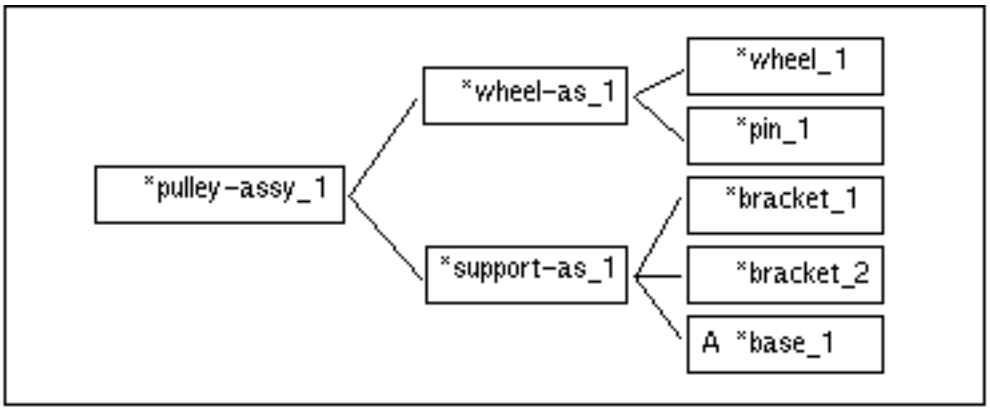
Figure 8-2 The Finished Pulley Assembly



Activating the Model Associated with the Base Component

Activate the model associated with the component. ACTIVATE MODEL creates the model associated with the `base_1` component.

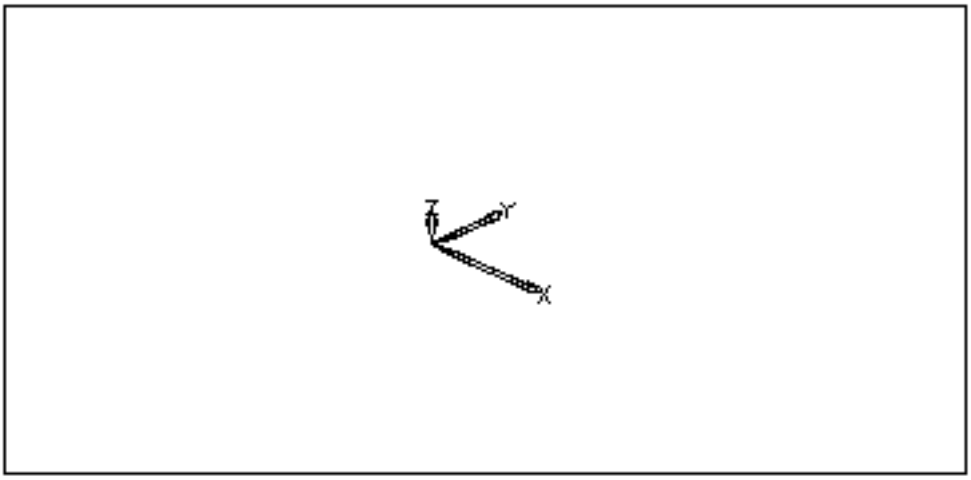
Figure 8-3 Pulley Assembly Structure after Activating the Model for base_1



Model Space Origin Location of the New Model

Activating a new model places the MSO location of the new model at the same MSO location of the assembly itself.

Figure 8-4 Graphics Window Showing Model Space Origin of the New Model for base_1



Building the Model Geometry

Create the basic geometry for the model associated with base, relative to the Right Cplane. All system-supplied Cplanes share the same X0Y0Z0 or MSO location.

Figure 8-5 Graphics Window Showing Basic Geometry of the New Model

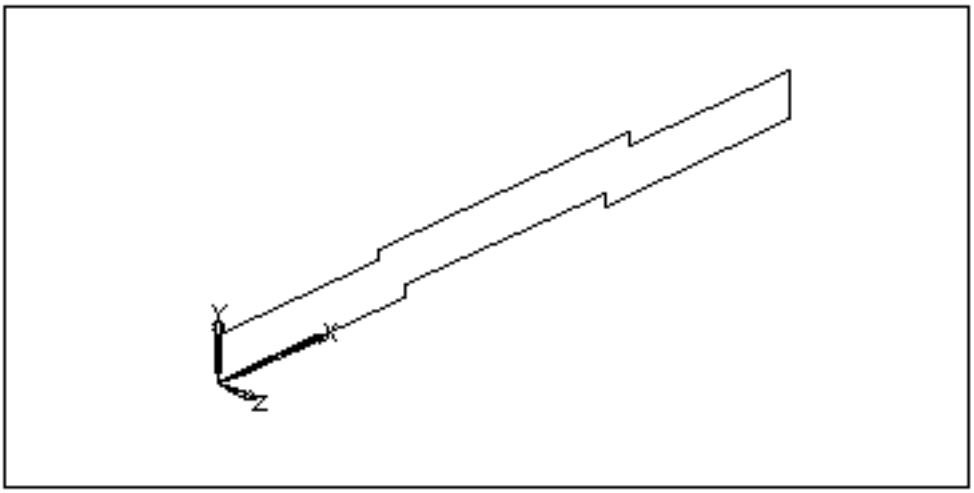
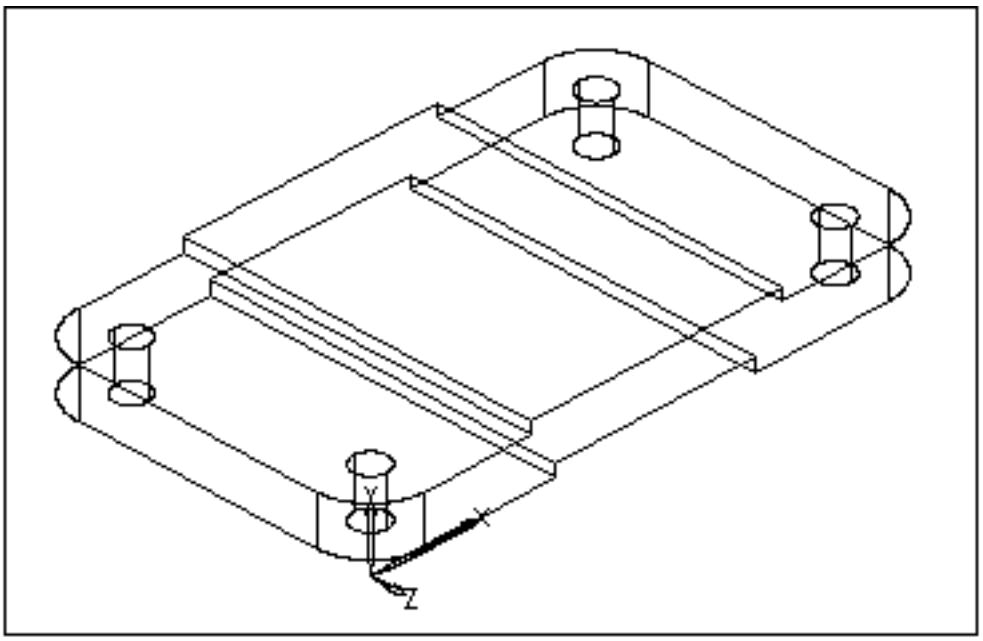


Figure 8-6 Graphics Window Showing Finished Geometry of the New Model for base_1



If you activate/create the model associated with `bracket`, and build its geometry relative to the assembly layout, the origin location of the new model is at the same Model Space Origin location of the assembly itself. Creating model geometry at the origin location of this particular assembly might be confusing to other users who use this model in their assembly. This is because the origin of the bracket is not at a location that other users can easily reference.

Figure 8-7 Assembly Structure after Activating a New Model for bracket_1

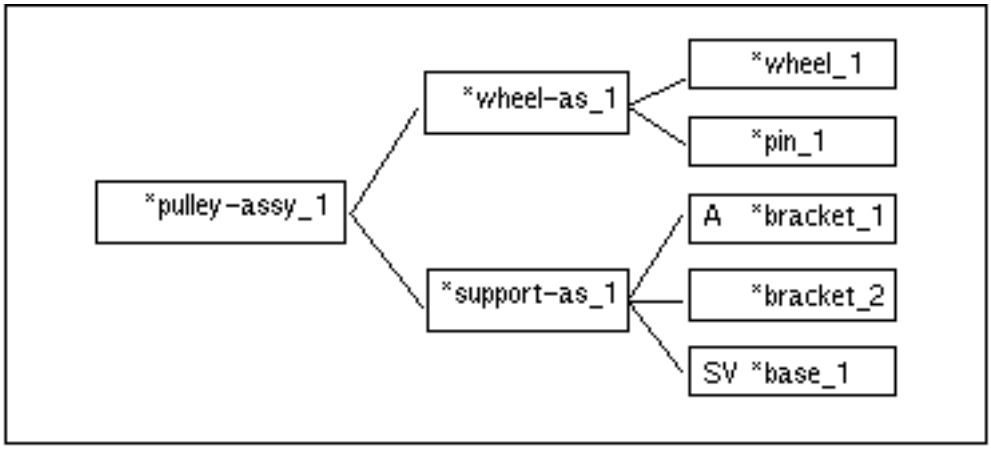
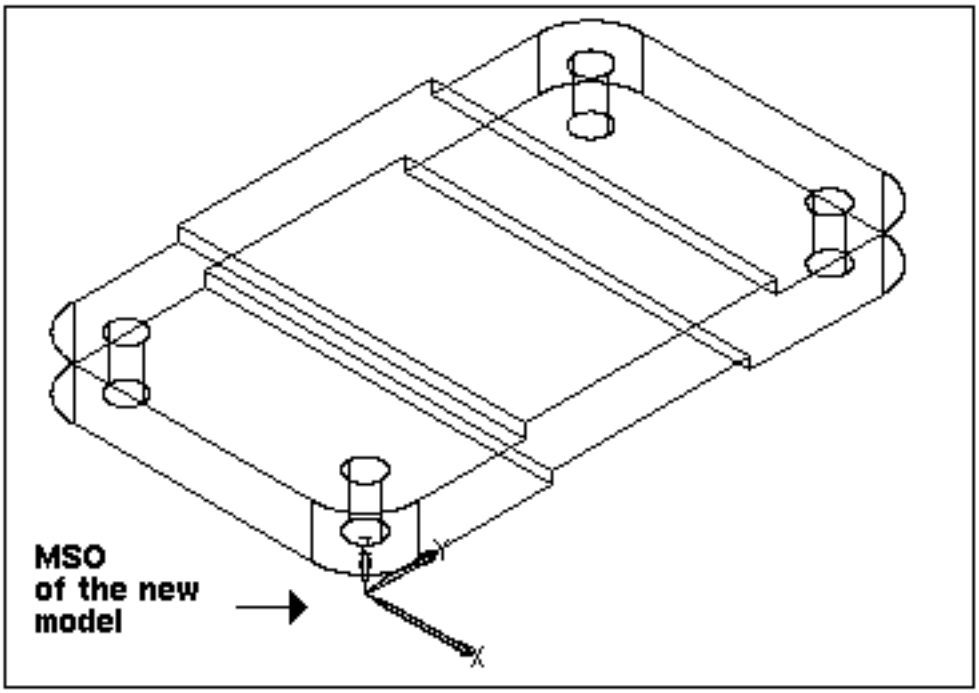


Figure 8-8 Undesirable MSO Location for the New Model Associated with bracket_1



The next section shows how to specify the MSO at a standard or easily identifiable location before creating a new model, so that other users can position the new model in other assemblies.

Specifying Model Space Origin before Creating the Model

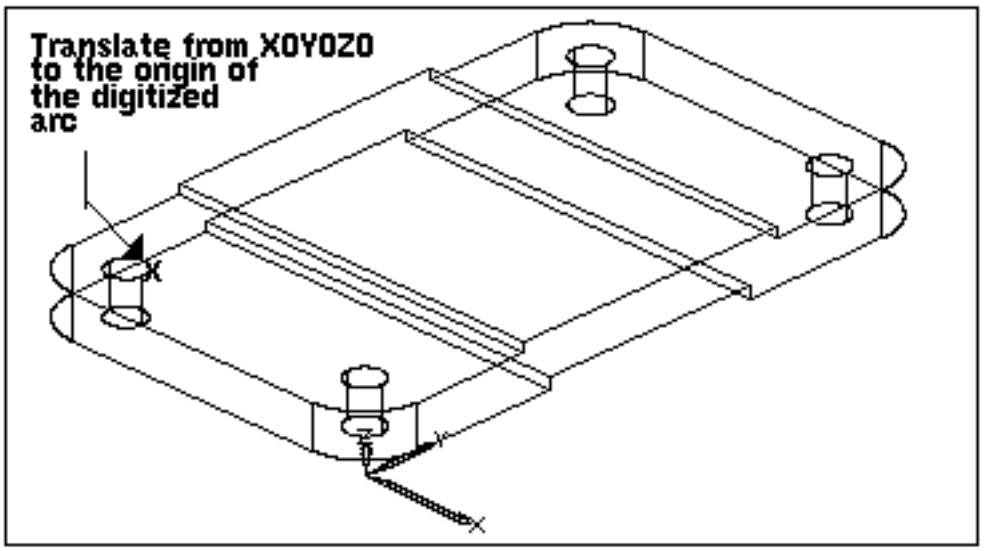
This procedure allows you to build the geometry of the new model in the context of the assembly layout, and locate the MSO of the new model at a standard location for use in other assemblies. This standard location is at the bottom of one of the counter-bore holes.

Position the Model Instance Associated with Bracket

Instead of activating the bracket component to create its model geometry, first locate the model instance using Move Component. Activate or create the model. Because the model does not exist for the bracket component instance, the handlepart is used to position the model instance in the assembly.

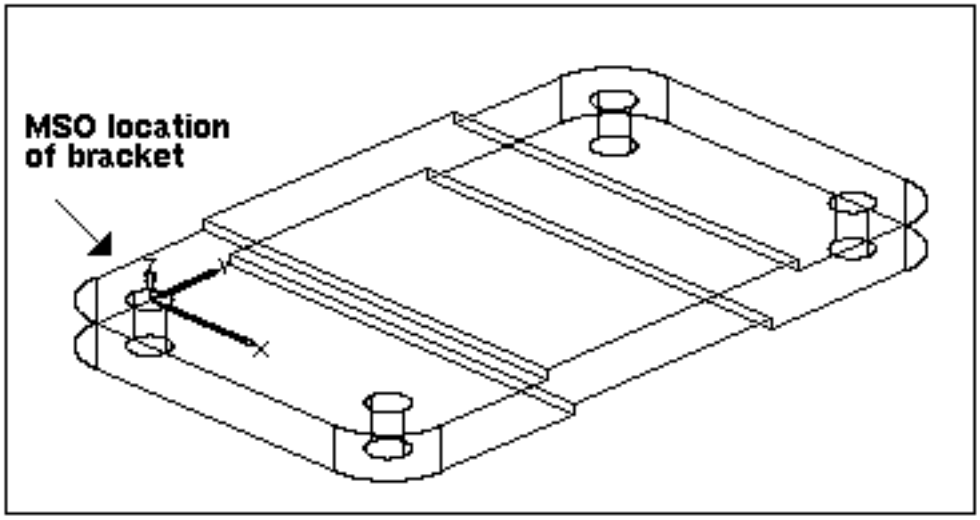
1. Select the `bracket_1` component instance in the Assembly Structure window. Choose the Move Component option from the pop-up menu to display the Move Component property sheet.
2. Select the Translate icon on the Move Component property sheet to move and position the bracket component instance to a specified MSO location.

Figure 8-9 Specifying Model Space Origin for the New Model associated with bracket_1



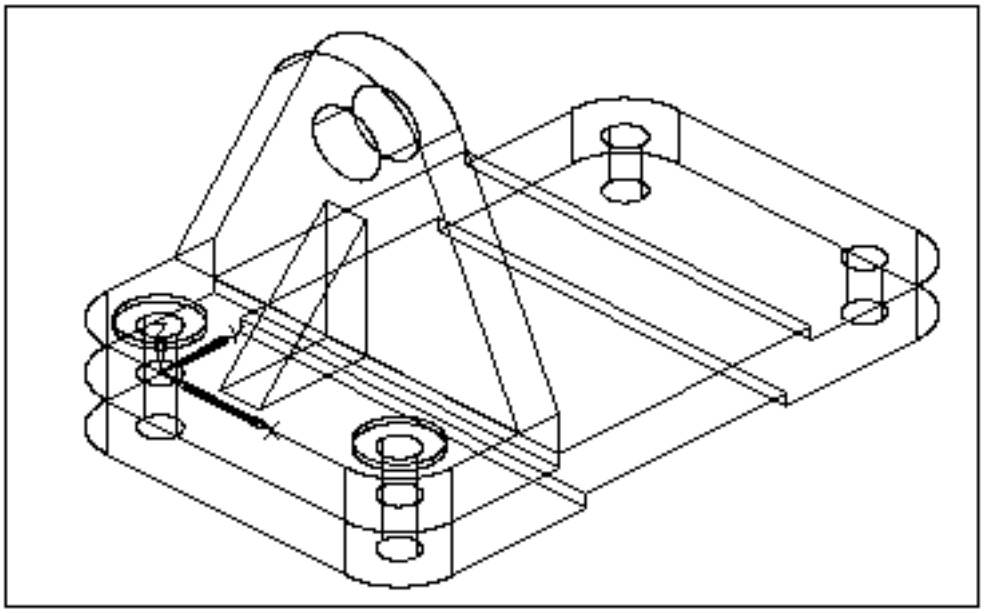
As the model itself does not exist after the Translate command is processed, there is no visual indication that you've moved the component instance or that the MSO of the model resides at the location you just specified.

Figure 8-10 Model Space Origin Properly Located for the Model associated with bracket _1



3. Use the handlepart graphics to check the placement of the model instance. The following is the completed model geometry created in the context of the pulley assembly.

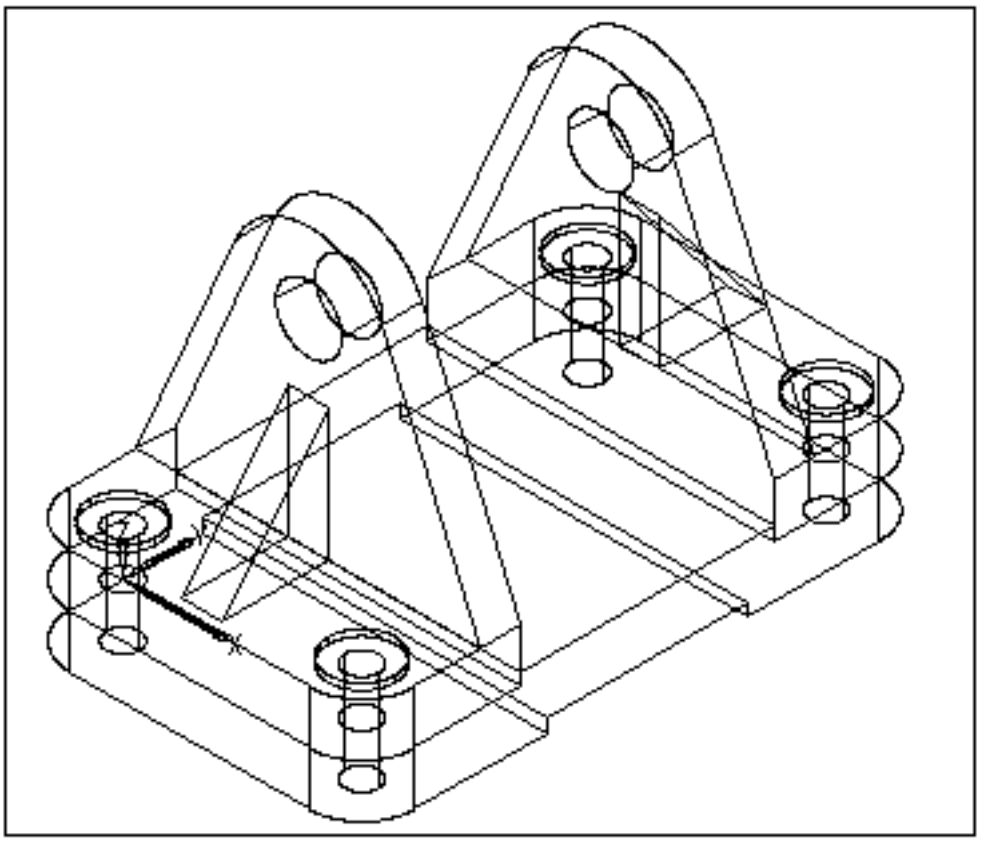
Figure 8-11 Geometry Completed for the Model associated with bracket_1



4. After specifying the MSO of the new model, activate the model associated with the component instance and create its geometry in the context of the assembly layout.

The second bracket instance is properly located and oriented in the assembly. The following example shows how to locate MSO for the new model associated with `wheel`.

Figure 8-12 Second Bracket Model Instance Properly Positioned in the Pulley Assembly

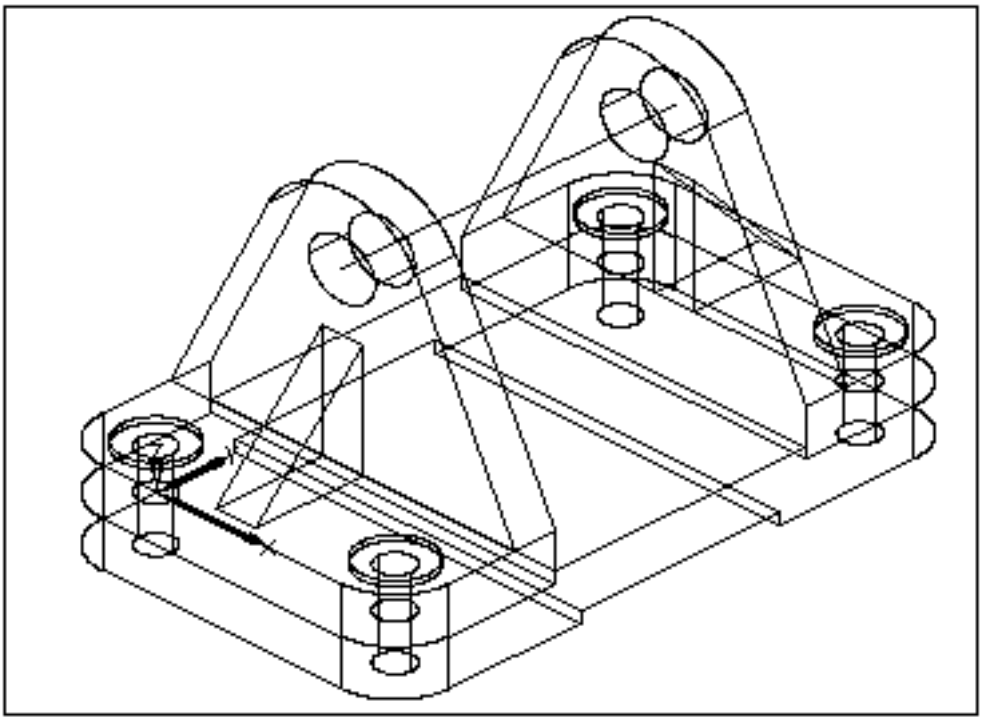


Locating MSO for New Model Associated with Wheel

Follow this procedure to locate the MSO for the new model associated with wheel.

1. With both bracket model instances properly positioned in the assembly, insert a construction wireframe line between the two bracket instances. Use the construction line to specify the MSO for the model associated with wheel. The resulting figure is shown on the following page.

Figure 8-13 Pulley Assembly after Inserting the Construction Line



2. Select either the Translate icon or the Orient icon to position the wheel component instance. At the prompt, specify the midpoint or origin of the construction line as the MSO for the wheel component. Use either the Translate icon or the Orient icon to position the wheel. Since the model geometry does not yet exist for the wheel component, use the handlepart graphics to visually check the placement of the component.

Figure 8-14 Move Component Property Sheet

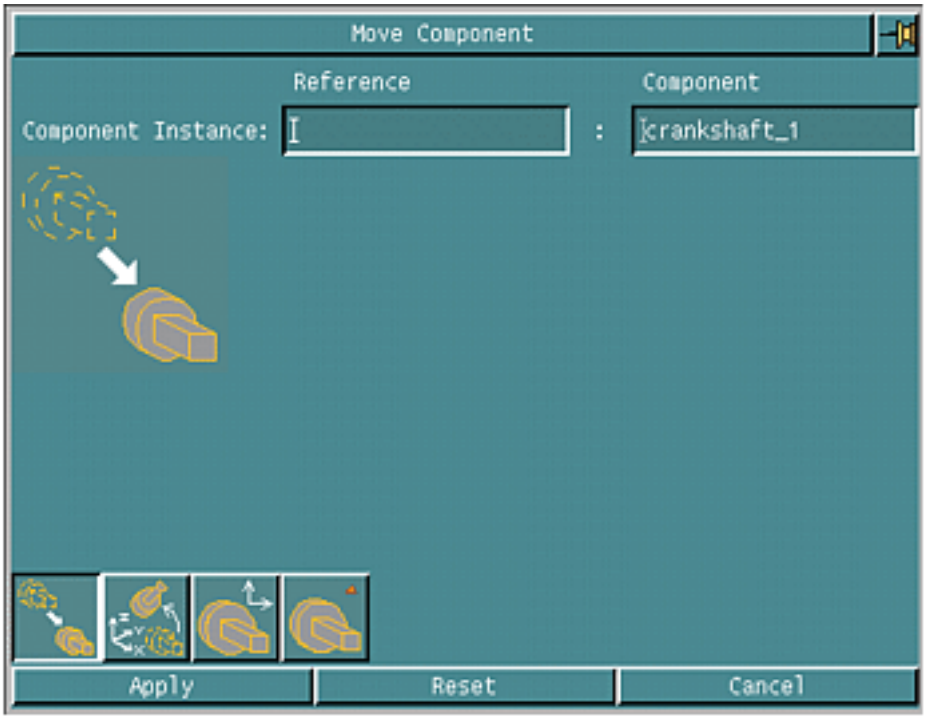
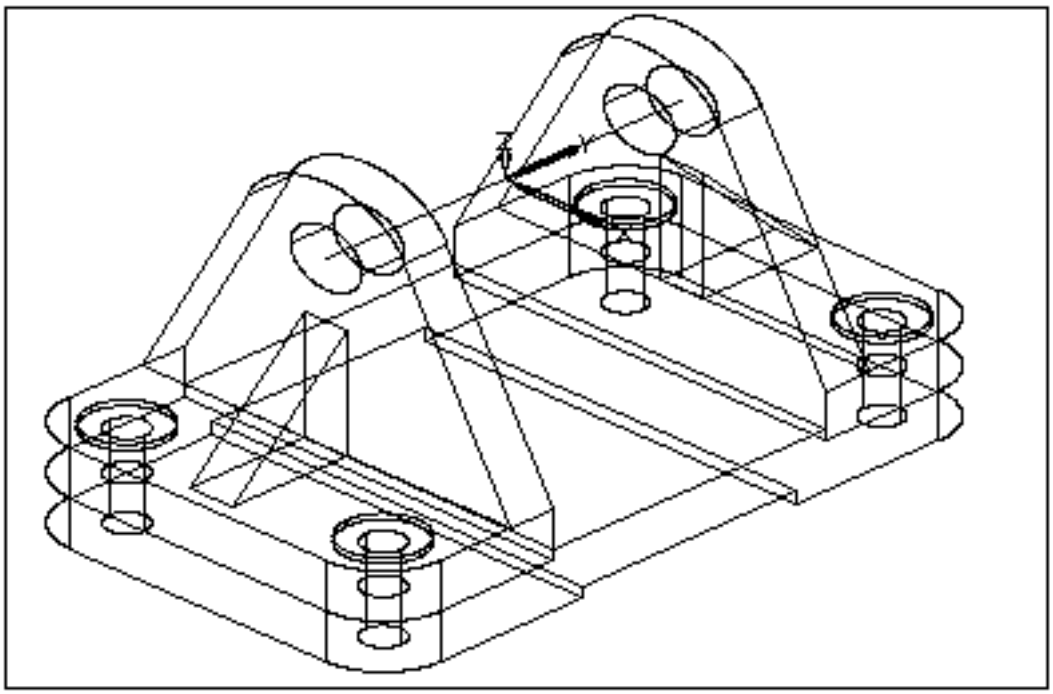


Figure 8-15 Locating Model Space Origin for the New Model Associated with Wheel



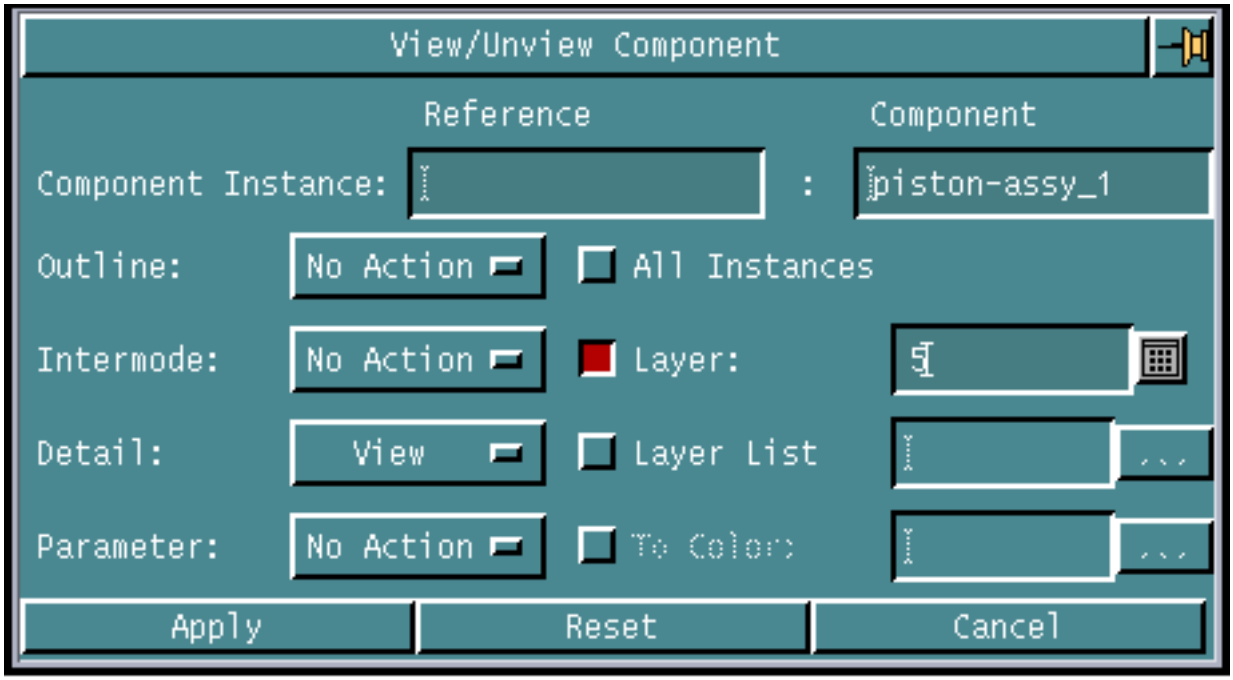
3. After the MSO is specified for the wheel component, you can activate the model associated with the wheel component to create its geometry.

Viewing a Component's Associated Model Geometry

When working in an active Adrawing, use View Component to see a component's associated model geometry.

1. Select the component you want to view from the Assembly Structure window.
2. Choose the View Component option from the pop-up menu. Or, select COMP/ASSY from the top bar and choose View from the menu that appears.

Figure 8-16 View/Unview Component Property Sheet



3. Specify the Component Instance as indicated below:

- Enter the name of a component instance in the Component Instance field in order to view the model associated with it or the models associated with its descendants.
- Enter the name of an assembly instance in the Reference field to view the models of a reference assembly based on a Component in that reference assembly.

Each assembly drawing (Adrawing) can have a unique layout of viewed and/or unviewed components. Some components might be viewed off and some might be viewed on for different layers. Layer lists are also unique for each Adrawing.

- Determine which components you would like to see on the assembly drawing.
- Decide whether or not to view the leaf components.
- Decide on which layer you wish to view the leaf components.
- Decide which, if any, layer ranges you would like to use for specific component or assembly instances.

You do not need to lock a component to view it.

Please note: Layer viewing conditions established for a component by the VIEW COMPONENT command are lost if you file the assembly with that component unviewed. To avoid this, create an execute (Explicit) or command (Parametric) file that captures your VIEW COMPONENT commands.

Options on the Property Sheet

These options are available for viewing the component.

- Outline - Allows you to view, unview, or make no viewing change to the model associated with the specified component instance (see "Outline Geometry" on page 8-20).
- Intermode - Allows you to view, unview, or make no viewing change to the components between the selected component and the leaf components.
- Detail - Allows you to view, unview, or make no viewing change to the model(s) associated with the leaf node descendants of the specified component instance.
- Parameter - Allows you to view, unview, or make no viewing changes to the parameters that were displayed the last time the associated model was filed.
- All Instances - Allows you to view all instances of the selected component. By default, it is off, which allows viewing of the single component instance.
- Layer - Allows you to specify a value by which to increment layer display. For example, if the model geometry resides on layers 3 and 30 and you specify a Layer value of 6, the model geometry appears on layers 9 and 36 in the active Adrawing. The default value is 0.

Please note: When you specify the layer number in the Parametric environment, the component is viewed on a layer which is the sum of the specified layer and the existing part layer. For example, if the component is on Layer 2 of a part, and you specify layer 23 in the assembly, the component is viewed on Layer 25.

- Layer List - Allows you to specify a range of viewing layers for the associated model geometry. For example, the associated model may contain geometry on layers 2 through 5 and layers 10-15 that you wish to see, while you do not wish to see the geometry that resides on the other layers in the model. Separate individual layers and layer lists by a comma. The default range is 0-254, which is all layers.
- To Color - Specifies a color for the component. A valid color number ranges from 0 to 65. Zero resets the color component status.

Outline Geometry

You can create outline, additional layout, or envelope geometry in a model associated with a subassembly component (non-leaf node). This is a useful way of showing the space that a subassembly requires in the assembly, without showing all the detail of the piece parts in that subassembly. It can be used to simplify the display of the component's leaf nodes (or child components).

If you want to use this technique, activate a model associated with a subassembly node and build the geometry.

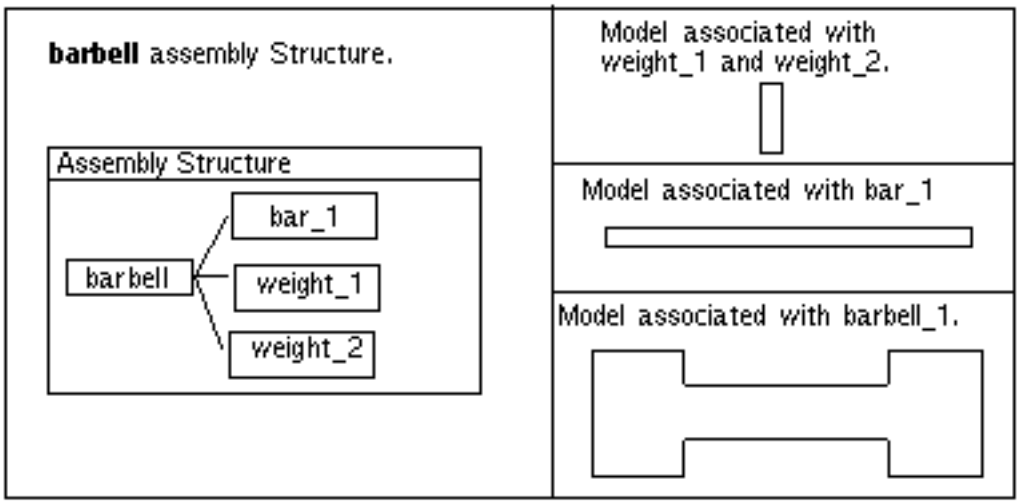
Example of Using Outline Geometry Technique: The examples that follow use a `barbell` assembly to illustrate how to display outline and/or detail geometry using the View/Unview Component property sheet. The first figure shows the `barbell` assembly and the geometry associated with its components.

The figures show the assembly with the results of specifying the following on the View/Unview Component property sheet:

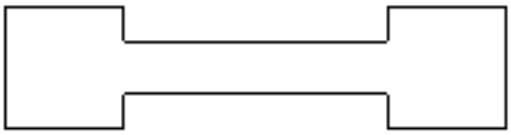
- Outline: View and Detail: Unview
- Outline: Unview and Detail: View
- Outline: View and Detail: View

The left side of the figure shows the `barbell` assembly structure. The right side shows the graphics for each component's associated model. (These figures are not an accurate representation of the CADDS Graphics window.)

Figure 8-17 barbell Assembly and Associated Model Geometry

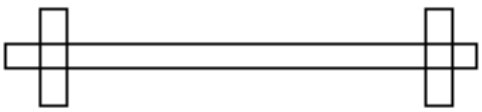


Select Outline: View and Detail: Unview on the View Component property sheet. Geometry associated with the specified component appears in the Graphics window.

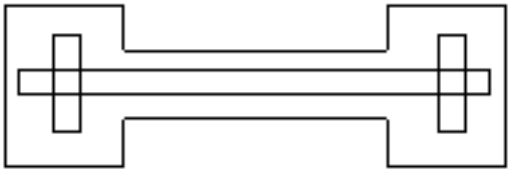


In this example only geometry associated with the `barbell_1` component is viewed.

Select Outline: Unview and Detail: View on the View Component property sheet. The geometry associated with the leaf components of the specified component appears in the Graphics window. In the example, geometry associated with the `bar_1`, `weight_1`, and `weight_2` components (that is, `barbell_1`'s leaf components) are viewed.



Select Outline: View and Detail: View on the View Component property sheet. The geometry associated with the specified component and its leaf components appear in the Graphics window.



In the example, geometry associated with the `barbell_1` component and the geometry associated with the `bar_1`, `weight_1` and `weight_2` components is viewed.

Controlling Layer Display

In both the Parametric and Explicit assembly environments, you can show model graphics with true layer differentiation. For example, if you have used separate layers to house parameters, model geometry, and so forth. CAMU maintains layer separation when viewing the model geometry of component instances.

Using the Layer option: You can use the Layer option to view geometry on either the layer(s) that the geometry resides on in the associated model (using a Layer value of 0), or you can specify a Layer value to use in incrementing the layer display. For example, if you use a Layer value of 2 when viewing the associated model of a particular component instance, all geometry in the model is incremented by a value of 2 when it appears in the active Adrawing (see "Options on the Property Sheet" on page 8-19).

- Specifying a Layer value
- You can use the Layer field to specify the value with which to increment the actual layer value(s) of geometry in the active part.This enables you to view the model graphics on a layer different from the layer(s) on which the CADDS model graphics reside. The geometry in that part itself is not affected.
- Not specifying a Layer value or specifying a value of 0
- Model graphics view on the same layer(s) on which they reside in the actual CADDS model, regardless of the active layer of the active Adrawing in which you are currently working.

Using the Layer List option: You can use the Layer List option to view only selected layer ranges (for the component or assembly instance specified) in the active Adrawing (see "Options on the Property Sheet" on page 8-19).

- Specifying a layer list enables you to control which aspects of the associated model geometry are visible in the Adrawing, based on their layer residency in the model.
- Not specifying a layer list defaults to a range of 0-254, which is all layers.

Temporarily Blanking Viewed Model Geometry

If you Unview a component while active in an Adrawing and then file and exit the assembly, its Layer and Layer List settings are lost for that Adrawing. To avoid this, do one of the following:

- Use Blank Component to temporarily view off the model graphics in the Graphics window. This retains the component's viewed layer assignment in the Adrawing. The blanked state is not saved when you file the assembly.
- Create a procedure file (Parametric) or execute file (Explicit) that captures the applicable ACTIVATE ADRAWING and VIEW COMPONENT command . Have it available for recreating the Layer and Layerlist settings.

Checking part database of a Model Associated with the Component

You can check the part database of a component while activating the model associated with the component, or viewing the component, using a third party executable ckCAD. You can use the ckCAD executable by setting an environment variable DB_CHECKING in the .caddsrc_local file (refer ".caddsrc-local File Information" on page 2-3).

If the part database of a component is corrupt, you cannot view, activate, or file (save) that component. Once the part associated with a component is identified as corrupt, the component and all its instances are marked with character "F" in the Assembly Structure. Any further attempt to view or activate any instance of the component is denied, as long as it is marked with character "F". If you try to view/activate this corrupt component, a message appears, indicating that the component is corrupt.

The character "F" on the component instance is not permanent and is valid only for the active session of the assembly.

Use the Change Component option to change the associated model part of the component. The changed part, if not found corrupt, is not marked with the "F" character. This allows you to view the model part of the component instance.

Moving a Component

Options from the Move Component icon provide ways to translate, rotate, orient, or position components in the assembly.

Select the Move Component icon from the ASSEMBLY/PAR or ASSEMBLY/EXP task set. The property sheet displays four icons for moving a component instance in the context of the assembly.

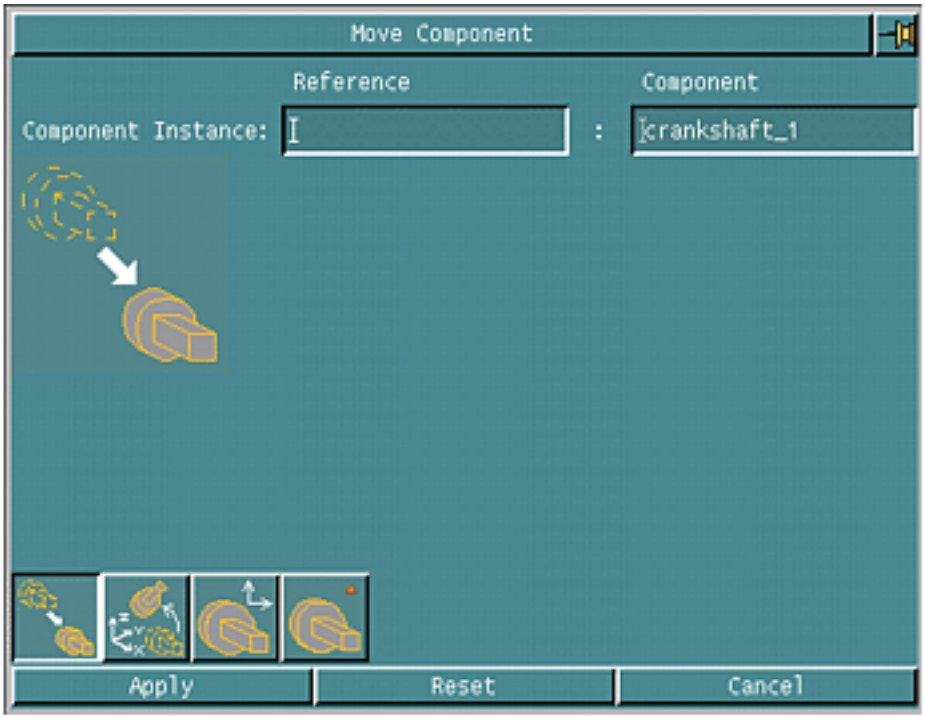
The component and its parent component must be locked. You do not have to lock the associated model, because you are not changing the model's geometry. You are changing only the location of the component instance in the assembly.

Please note: You can use the Lock Component function to lock a desired component. (see "Determining What to Lock" on page 5-32).

Please note: If you have issued Select Assemble On (see "Automatic Assembly" on page 6-32) all parametric positioning constraints are solved when you use any of the Move Component property sheet options.

Select a component instance from the assembly structure or from the viewed model geometry.

Figure 8-18 Move Component property sheet



Please note: The Move Component property sheet differs somewhat between parametric and explicit environments. However, the functionality is essentially the same.

Using Translate, Rotate, and Orient Component

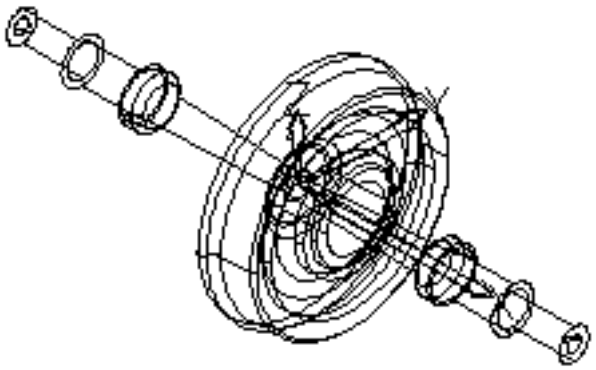
In this procedure, the term component refers to the associated model geometry in the Graphics window.

To move a component from one location to another, perform the following steps:

1. Activate the assembly. In the example, the user activates the assembly `wheel-assy` in the Explicit environment.
2. Activate an Adrawing. Define one or more views.

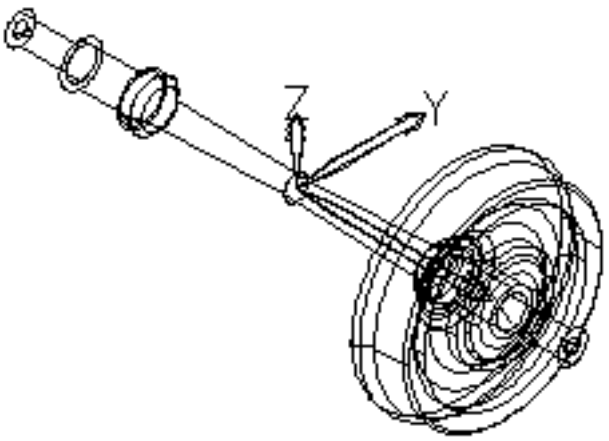
   In the example, the user selected ISO view orientation, view scale .5, set view.
3. View the components `axle_1` and `wheel_1`.

Since the View and Position option was not selected when the components were added to the assembly, these components appear at their original locations.

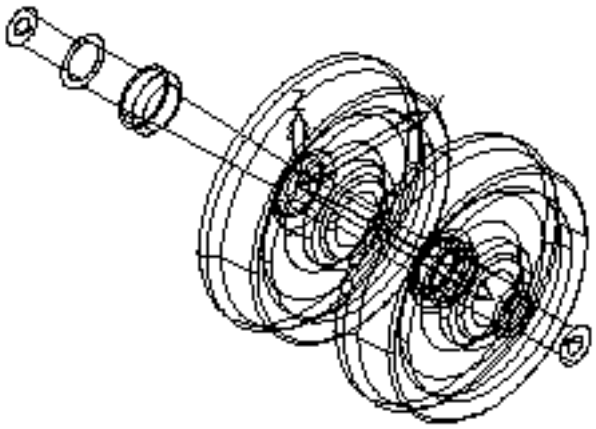


4. Choose the Move Component option from the COMP/ASSY task set. The Move Component property sheet appears.
5. In the Component Instance field, type in the component name `wheel_1`.
6. Click Apply.
7. At the prompt, specify a model location, for example:
   `Model loc X0Y0Z0 ORG d1`

   The figure below shows the `wheel_1` component after translation.



8. View the component `wheel_2`. The component is viewed at its default MSO location.



To properly locate `wheel_2` you can also use the Orient Component option (see "Alternative Procedure for Orienting and Positioning wheel_2" on page 8-28).

9. Select the Move Component and Rotate component icons from the property sheet.

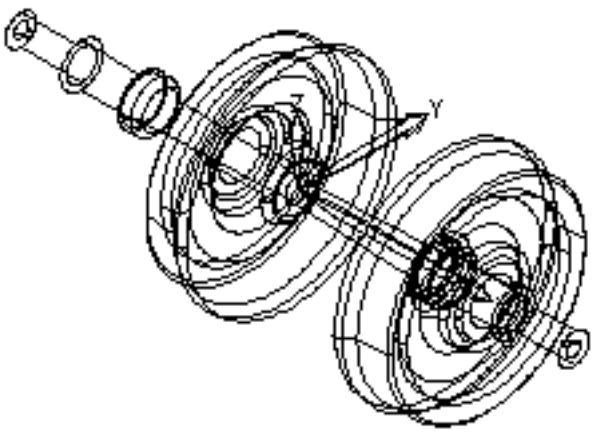
Click the AZ button and type 180. This rotates the `wheel_2` component 180° around the Z axis. Click Apply. A prompt appears for you to define the center of rotation.

```
AZ 180: MODEL loc
```

10. Enter orientation modifiers or freehand locate to define model location. In the example, you can enter the following at the prompt:

```
Az 180: MODEL loc X0Y0Z0
```

11. In the figure below, `wheel_2` has rotated 180 degrees about the Z axis using X0Y0Z0 (its initial origin position) as the center of rotation.

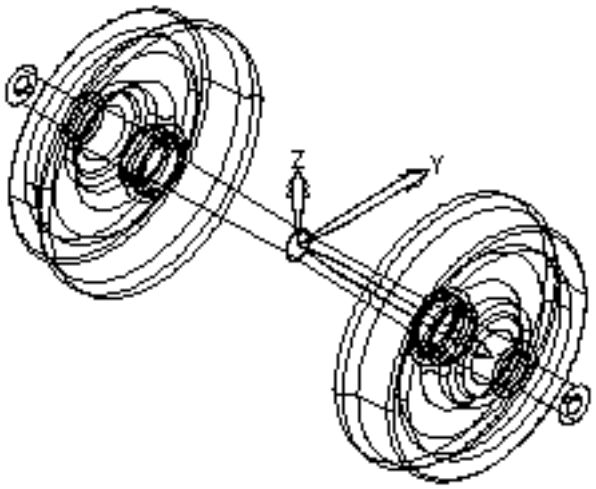


12. To properly locate `wheel_2` along the `axle`, choose the Move Component option from the COMP/ASSY task set. Select the Translate Component icon from the property sheet.

13. Click Apply. A prompt appears for you to specify a location.

    Specify `IX - 1.5875`

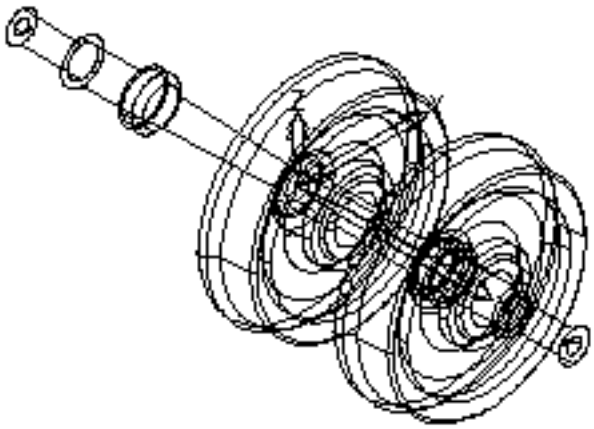
    The figure shows `wheel_2` properly located.



Alternative Procedure for Orienting and Positioning wheel_2

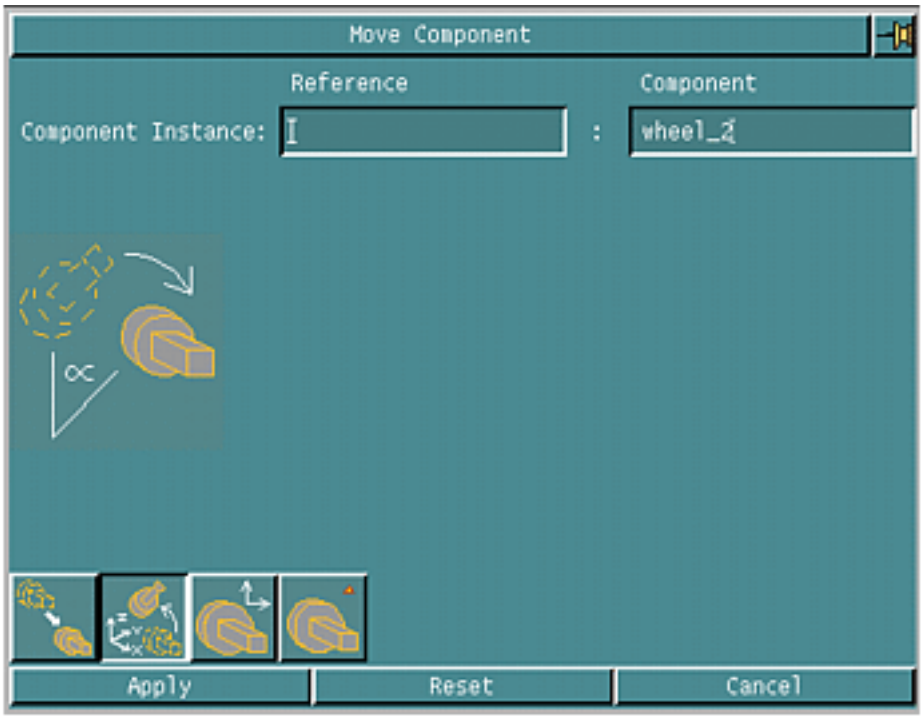
This procedure replaces Steps 4 through 12 in the “Using Translate, Rotate, and Orient Component” on page 8-25 example.

In this example, `wheel_1`, `wheel_2`, and `axle_1` are viewed. `wheel_2` is at its origin position, and must be correctly oriented and located.



1. Select the Orient Component option from the Move Component property sheet. Enter `wheel_2` as the name of the component instance in the Component Instance field.

Figure 8-19 Move Component property sheet with Orient option



2. Select AZ and enter 180.

   If you select more than one (of AX, AY, or AZ), the system rotates about the x axis first, then y, then z.

3. The prompt appears.

   `MODEL loc`

4. Enter a two-point translation vector. The second point defines the center of rotation.

   X0Y0Z0 IX -1.5875

These four steps simultaneously locate and rotate `wheel_2`.

To rotate about a second vector, enter a comma (,) and enter another pair of coordinates. You can use location filters (such as END, ORIGIN, INTERSECTION, and so on) and select viewed geometry.

Six-Point Orientation Method

The six-point orientation method is available in Explicit only.

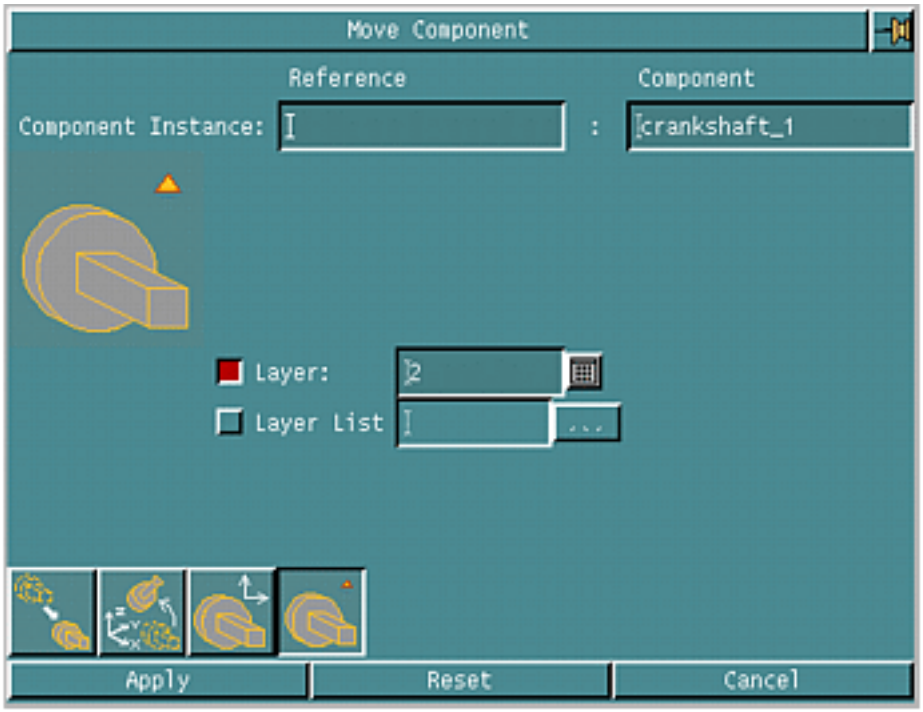
Position Component

Position Component allows you to view and position a component instance in the same procedure. Use Position Component when your component instance is not viewed. Position a component when you add it to the assembly, or later select the Position Component icon from the Move component property sheet.

To position a component using the Move Component property sheet:

1. Select the Position Component icon from the Move Component property sheet. The Position Component icon is displayed.

Figure 8-20 Move Component property sheet with Position option



2. Enter the name of the component instance or reference assembly instance you want to position in the Component Instance or the Assembly Instance field.
3. Click Apply. The Apply field changes to Abort, and the following menu appears:

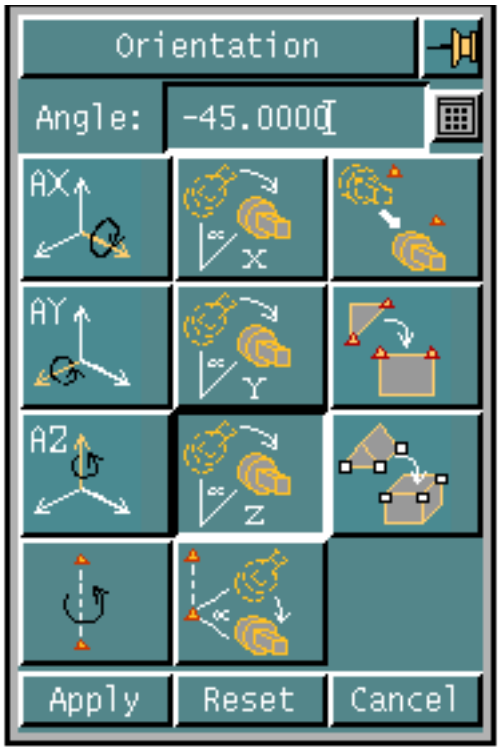
Abort
Orientation
Done

4. Click Orientation. An icon menu appears, which provides options for orienting a component.

Orientation Procedure

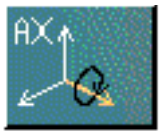
The following figure shows the Orientation menu with a brief description of each option. Information on the use of each option follows.

Figure 8-21 Orientation Menu



Rotating a Component around an Axis passing through the Origin of the Part

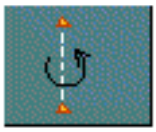
Use the following options to rotate a component around x-, y-, and z- axes or an axis defined using two locations.



Rotates the component around the x-axis passing through the origin of the component.

Rotates the component around the y-axis passing through the origin of the component.

Rotates the component around the z-axis passing through the origin of the component.



Rotates the component around an axis passing through the origin of the component and parallel to an axis defined by any two locations you specify.

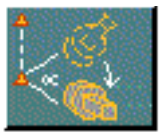
Rotating a Component around an Axis passing through a Selected Location

Use the following options to rotate a component around the x-, y-, and z- axes passing through a selected location or around an axis defined using two locations.

Rotates the component around the x-axis passing through the selected location.

Rotates the component around the y-axis passing through the selected location.

Rotates the component around the z-axis passing through the selected location.

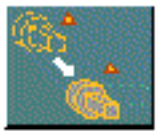


Rotates the component around an axis defined by any two locations. The rotation vector is taken to be along the axis directed from first location to the second.

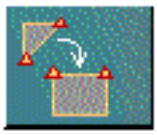
On specifying a rotation for a component, the rotation is done immediately. You can see the component rotate. When you specify a location for the part, you can see the result of the rotation.You need not specify a location as the component is already located.In case the component is not located, the component rotates and remains unviewed.

You can also rotate incrementally. Each rotation specified is added to the component’s current rotation about that axis.

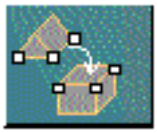
Translating, Aligning, and Adjusting Components



Translates the component by a displacement defined by specifying two locations.



Aligns the components vector by vector. Also rotates or translates the component.



Adjusts the components using two planes, each defined by three locations. Also translates and/or rotates the components along x-, y- and z- axes.

The RESET option resets the component's orientation to the last orientation of the previous instance.

Positioning tools include:

- ANGLE <angle> AXIS <axis>!
  Specifies the angle of rotation and the axis of rotation

  AXIS - defines the axis:

  - With two location points (FREE)
  - Along a selected line (LINE)
  - Through a specified point and parallel to the positive x-, y-, or z- axis of the active Cplane (X, Y, Z)

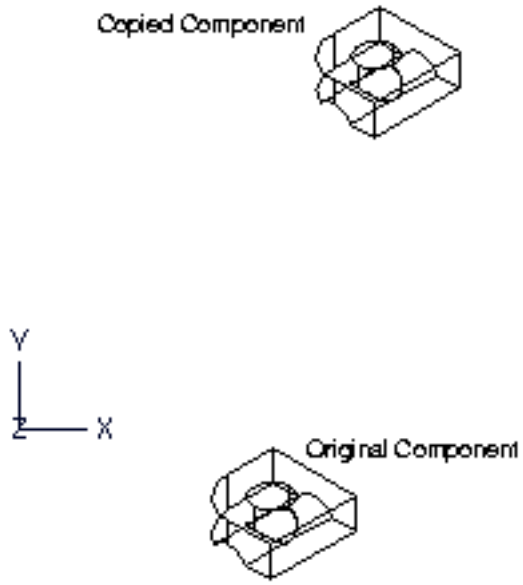
Orienting a Component: Work Examples

You can view and position the component after copying, moving or mirroring using the appropriate orientation options from the Orientation menu.

Explained below are a few examples.

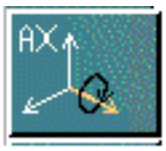
Following is the result of a Copy Component command. In this example the Cplane is ISO and is viewed in ISO view. The resultant component is positioned at location [50,50,0].

Figure 8-22 The copied component after positioning at a location but before orientation.



The copied component can be oriented as follows:

Example 1: Rotating a Component at its Origin around x-Axis

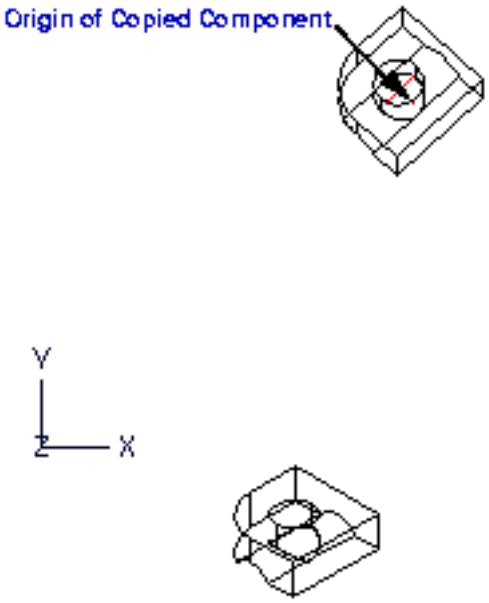


Rotates the component at its origin, around the x- axis with a specified angle of rotation.

- Choose the appropriate option from the Orientation menu.
- Specify 30 degrees as the angle of rotation in the Angle field.
- Click Apply.

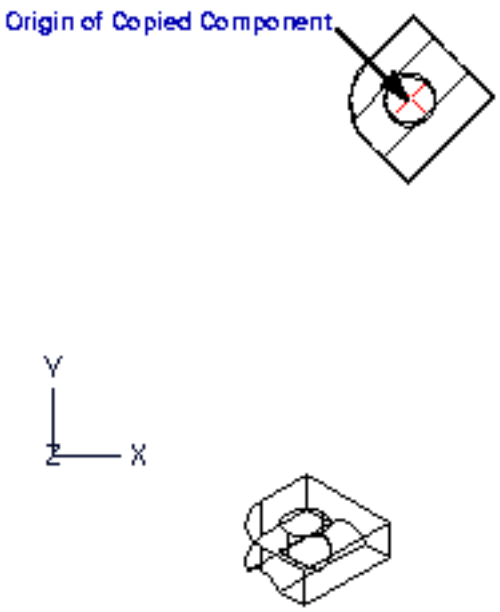
  The resultant geometry is rotated by 30 degrees at the origin of the copied component around the x- axis.

Figure 8-23 The copied component rotated at its origin along x- axis by an angle of 30 degrees.



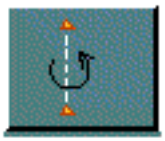
Rotating Incrementally: To rotate the component again by an angle of 30 degrees around the same (x-) axis, click the Apply option once more on the Orientation menu. The following is the resultant geometry.

Figure 8-24 The copied component rotated incrementally at its origin along x- axis by an angle of 30 degrees.



Follow the same procedure to rotate the component around the y- and z- axis about point of origin.

Example 2: Rotating a Component at its Origin around an Axis Defined by Specifying Two Locations

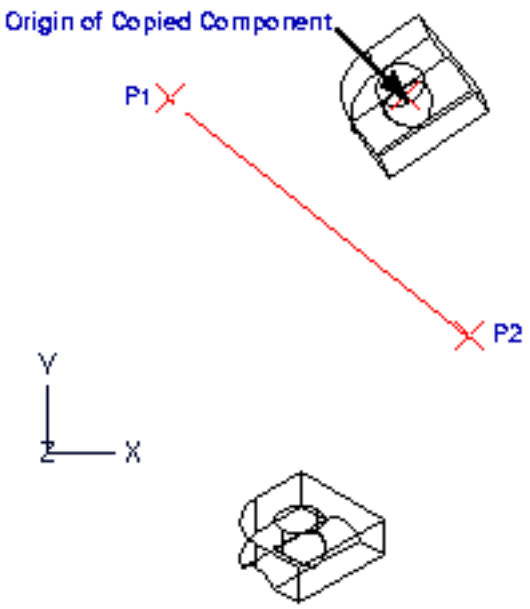


Rotates the component about its origin, around an axis passing through its origin and parallel to an axis defined by specifying two locations and the angle of rotation.

To rotate a copied component (refer to Figure 8-22) about its origin,

- Choose the appropriate option from the Orientation menu.
- Specify 30 degrees as the angle of rotation in the Angle field.
- Click Apply.
- Select two points, P1 and P2, to define the axis parallel to the required axis of rotation.

Figure 8-25 The copied component rotated at its origin parallel to an axis defined by points P1 and P2, by an angle of 30 degrees.



Example 3: Rotating a Component around the y-Axis passing through a Location

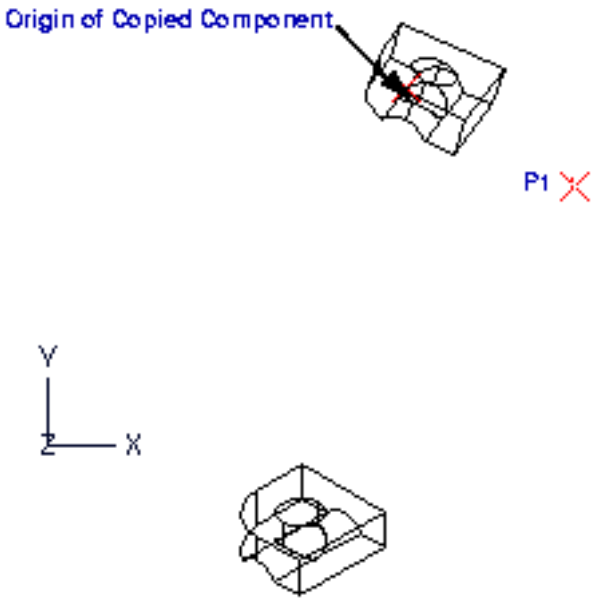
Rotates the component around the y- axis passing through a specified location.

To rotate the copied component (refer to Figure 8-22) around the y-axis passing through a location specified by point P1,

- Choose the appropriate option from the Orientation menu.
- Specify 30 degrees as the angle of rotation in the Angle field.
- Click Apply.
- Select the point P1 on the graphics area, to specify the location through which the axis is passing.

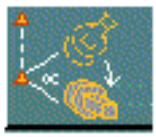
  The resultant geometry is rotated by an angle of 30 degrees around the y- axis passing through the location point P1.

Figure 8-26 The copied component is rotated around Y- axis passing through location P1 by an angle of 30 degrees.



Follow the same procedure to rotate the component around the x- and z- axis passing through a given location.

Example 4: Rotating a Component around an Axis Specified by Two Locations



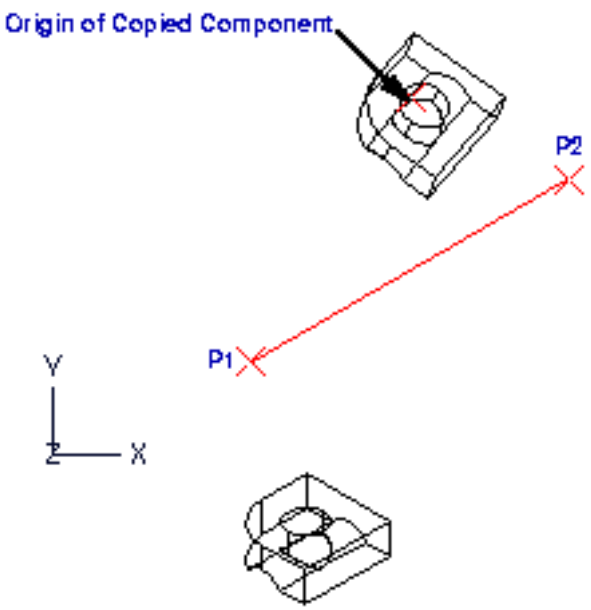
Rotates the component around an axis specified by two location points.

To rotate the copied component (refer Figure 8-22) around an axis defined by two location points P1 and P2,

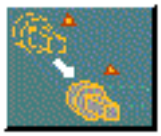
- Choose the appropriate option from the Orientation menu.
- Specify 30 degrees as the angle of rotation in the Angle field.
- Select any two points, P1 and P2 on the graphics area to define the axis of rotation.
- Click Apply.

  The resultant geometry is rotated by 30 degrees around the axis defined by location points P1 and P2.

Figure 8-27 The copied component is rotated around an axis defined by location points P1 and P2 by an angle of 30 degrees.



Example 5: Translating a Component by a Displacement Specified by Two Locations



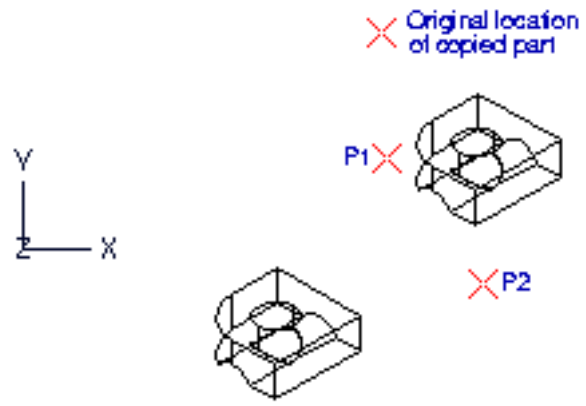
Translates the component by a displacement specified by two locations.

To translate the copied component (refer Figure 8-22) by a displacement specified by two points,

- Choose the appropriate option from the Orientation menu.
- Click Apply.
- Select a location on the graphics area to define the starting point P1 of the displacement vector.
- Select again a location to specify the end point P2 of the vector

  The resultant geometry is translated to the new location by the specified displacement.

Figure 8-28 The copied component is translated from the original location by a displacement specified by points P1 and P2.



Use the Align and Adjust options in a similar manner to,

- Align the copied component
- Adjust the copied component

Clearing a Component's Inherited Orientation

The following applies to the View and Position options of the Add Component and Copy Component property sheets, and options under the Move icon: when you add multiple instances of a component, the orientation specification can be inherited from a previously positioned component instance. Use the RESET option to clear an orientation specification inherited from a previously positioned component instance, or an unwanted rotation of the component you are currently positioning.

Positioning Tools

The options on Orientation menu provides to translate, align, and adjust. The bottom icon, RESET, resets to the last orientation of the previous instance.

- TRANSLATE <vector>!
  Translates the component using a vector. Specify the vector with two locations. This option translates the origin of the first location to the origin of the second location.
- ALIGN <loc><loc>^<loc><loc>^!
  Aligns the component you select with another component so that the vectors (defined by one set of locations on each component) align with each other. ALIGN conducts a translation and rotation of the component, allowing for possible translation in three axes and rotation in two axes.

  Coincident points — The system reorients the (first) component so that the first points selected on each component (1 and 3) are coincident.

  Alignment of axes — The system aligns the axes defined by the first and second points, respectively, on each component.
- ADJUST <loc><loc><loc>^<loc><loc><loc>^!
  Adjusts the plane of the selected component with the plane of another component (defined by two sets of three locations each) so that the first and fourth locations are coincident. It conducts a full rigid transformation of the component, allowing possible translation and rotation in all three axes. Requirement: Define the mating plane with three locations on the component you want to align, then provide three locations on the component with which you want to align the first component.

Utilities for Positioning

When adding multiple components, select a layer to view each component on a different layer.

- Layer Utilities

  You can view component model instances on different layers using Add and Position Component. Specify the layer before you locate the instance. If you do not specify a layer, the model geometry associated with the component instance views on the same layer on which it resides at the piece part level.
- Utility Construction Planes (Parametric)

  Create utility construction planes to use when you position components with the Add and Position commands.

  Using a utility Cplane has different results, depending on whether you select it before or after you locate or rotate a component.

- Select utility Cplane, then rotate or locate the component. The component is oriented with respect to the utility Cplane.
- Locate or rotate the component, then select utility Cplane. The component is oriented with respect to the active Cplane at the time it is located or rotated.

  Locate or rotate the component, then select utility Cplane. The component is oriented with respect to the active Cplane at the time it is located or rotated. The component does not use the utility Cplane for orientation, but uses it as a reference for any subsequent rotations and/or locations.

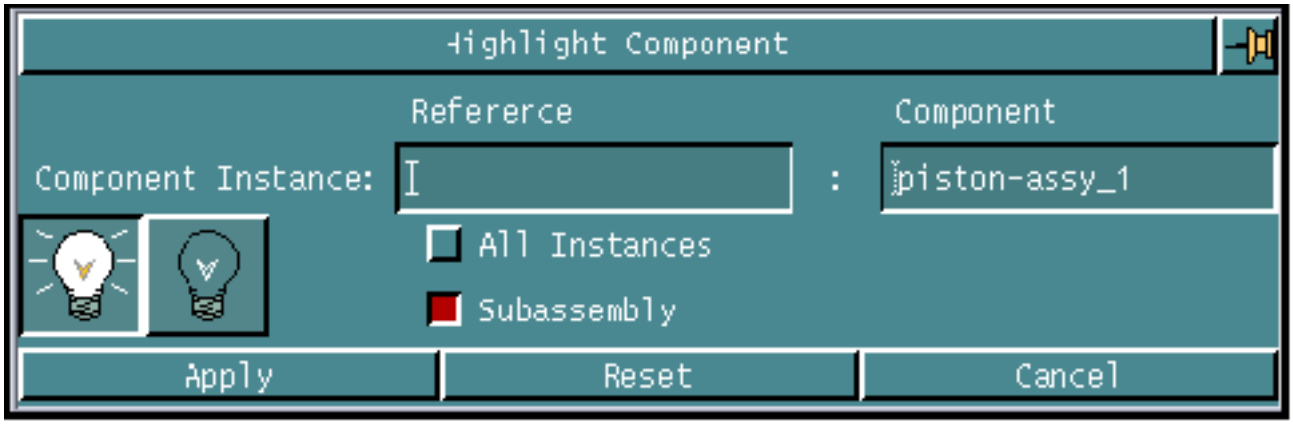
When you add multiple components, the orientation and selected Cplane (which could be a utility Cplane) are inherited by the next component instance you position.

- To override the inheritance of the orientation, select RESET to use the inherited Cplane.
- To override the inheritance of the Cplane, select a new utility Cplane, then RESET. This matches the orientation of the component to the new utility Cplane.

Highlighting the Appearance of Geometry

Highlight Component lets you highlight model geometry associated with a specific component or subassembly and save the highlighted view state upon filing the active Adrawing. The highlighted view state is retained the next time you activate the Adrawing. Components and their corresponding model geometry remain highlighted until you turn off the highlighting using the Highlight Component property sheet.

From the top bar, select COMP/ASSY. Select Highlight, or select the component you want to highlight and then Select Highlight Component from the Assembly Structure menu.



This technique is helpful if you wish to visually associate model graphics that correspond to a particular component or subassembly. Highlighted components display an “H” in the component box in the Assembly Structure window.

By default, components and their corresponding model geometry highlight automatically (display in reverse video) when you select the component from the Assembly Structure window. Conversely, selecting the edge of a viewed solid model in the Graphics window highlights the entire solid model and its corresponding component (displays in reverse video) in the Assembly Structure window. The default highlighting is not retained and is removed as soon as you turn off the selection of the component and/or its associated model graphics.

Chapter 9

Using Interpart Constraints

This chapter describes how to establish constraint relationships between different models in the same assembly. The topics are:

- About Interpart Constraints
- Selecting Variables from a Model to Include in Your Assembly
- Placing Component Variables in the Active Assembly
- Constraint Variable Names of Imported Components
- Preparing to Add Equations Between Variables
- Establishing Relationships Between Variables
- Changing Variable Values and Applying the Changes
- Regenerating the Component
- Forcing a Variable to Remain Constant
- Deleting a Variable

About Interpart Constraints

Constraints are a capability of the Parametric modeling environment. They allow you to define variables and write equations that define the relationships between variables in models.

When you design model geometry in the parametric environment of CADDS, the system creates parameters that define characteristics of your model. With constraints, you can define mathematical relationships between parameters. When one parameter is changed, the other is made to change based on the relationship you established. This is described fully in the *Parametric Modeling User Guide and Menu Reference.*

In the Parametric environment of Concurrent Assembly Mock-Up, you can establish constraint relationships between variables in different models in the same assembly. You define variables and establish their relationships using methods similar to those you would use from the CADDS Parametric modeling environment (without Concurrent Assembly Mock-Up). All the Constraints icons are available to you. However, the Constraints task set in CAMU has additional icons that allow you to constrain between models in your assembly.

Please note: Interpart constraints, described in this chapter, and parametric positioning, described in Chapter 6, “Parametric Positioning”, are two separate and distinct means of using parametrization in an assembly. They are two different systems.

Basic Steps to Use Constraints in an Assembly

The following steps give an overview of how to constrain between models in your active assembly.

1. Activate the assembly in the Parametric environment.
2. Activate a model and assign a variable to each parameter you want to relate to a parameter in another model of the assembly.
3. Export the variable(s) in the model you want to relate to parameters of another model in your assembly.
4. Activate the other model. Assign a variable to each parameter as you did in the first model.
5. Export the variable(s) of the second model that you want to relate to the variable(s) you exported from the first model.
6. To constrain a variable in one model to a variable in another model of the assembly, activate the assembly’s root node model.

7. Import the components representing the models that contain the variables you exported.
8. Write an equation to relate a variable from one model to a variable from another model.
9. If you change the value of one variable, solve the imported components (representing the models) that contain variables affected by the variable value change.
10. Regenerate the components whose associated models contain variables affected by the change.

The assembly can contain any number of constraint relationships among its associated models. You can export and import more than one variable at a time. You can also solve/regenerate as many components as you want at the same time by selecting several components on the property sheets.

To increase the number of models that can be suspended, set the following variable in your `.caddsrc-local` file, where *nn* is the number of models you want to suspend, plus 1 (for the active model). The default value is 6:

```
setenv CADDS_MAX_PART nn
```

Selecting Variables from a Model to Include in Your Assembly

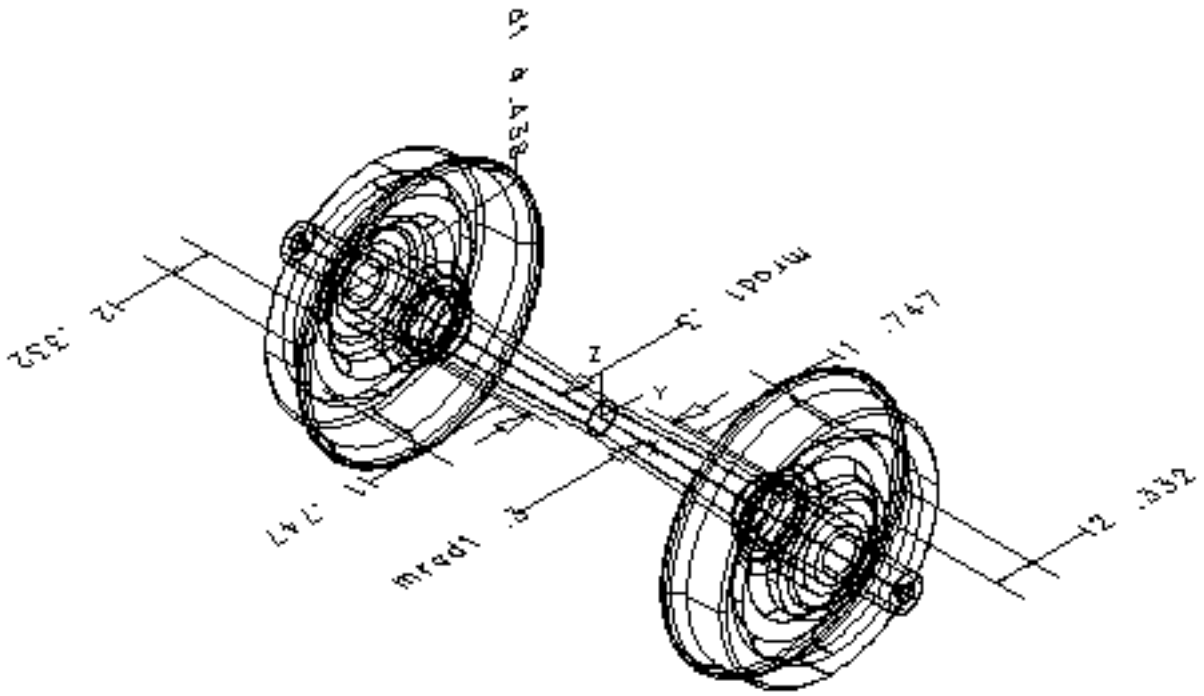
Use Export Variable to specify the variable(s) from each model you want to constrain. You can do this in the Parametric modeling environment or in the Parametric assembly environment.

Please note: It is best to constrain as many variables as possible in the component model to control design of the model. Export only the minimum amount of variables that are necessary to control constraints between components of the assembly.

This procedure assumes that you have already assigned variable names in your models and have defined relationships between variables in each model.

In the example that follows, the user is working in the assembly `wheel-assy`, with all components viewed in the isometric orientation. The user brings into the assembly variables from two different part models: The variable `mrad1` from the model `wheel` and the variable `d1` from the model `axle`.

Figure 9-1 wheel-assy

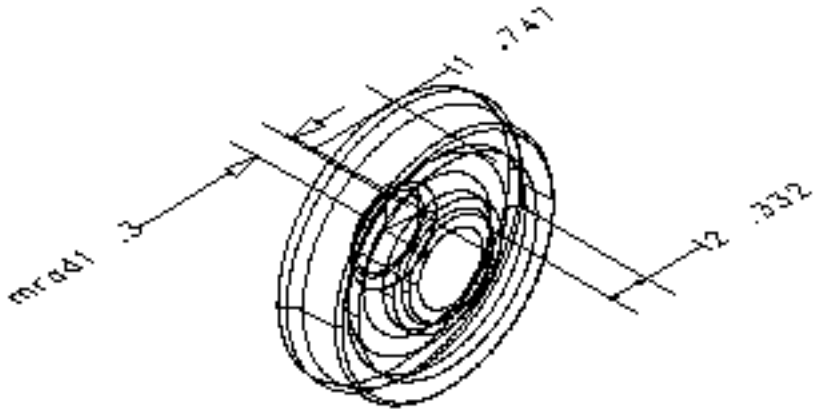


From each of the models between which you want to create constraints, use Export Variable to specify the variables. You can do this in the single part environment, or in CAMU.

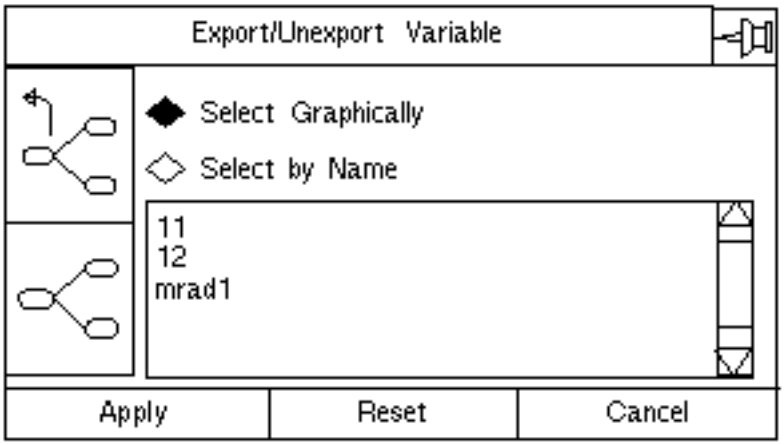
Follow these steps.

1. Activate the model that contains variable(s) you want to relate to variable(s) in another model of your assembly. Use the pop-up menu in the Assembly Structure window to select Activate Model.

   In the following example, the user has activated the model associated with the component `wheel_1`. Its parameters are visible.



2. Select the Export Variable icon. The Export Variable property sheet displays a list of variables associated with the model.



   In this example, the user selects the variable `mrad1`.

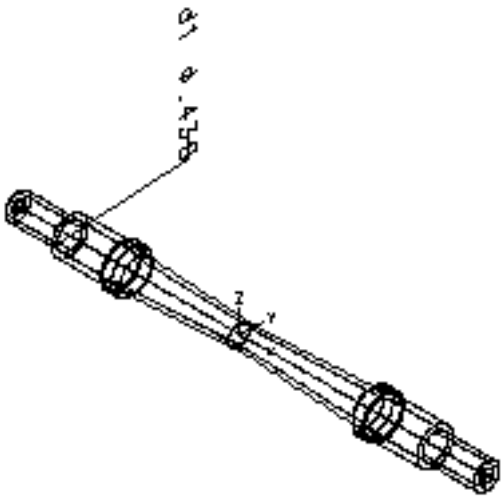
3. Select the variables to be used in the assembly.

4. Click Apply.

   If two variables are already constrained together, such that one variable determines the value of the other variable, export only one of these variables from the model. If you use both of these variables in another equation in the assembly, and the two constraint relationships contradict each other, the equation is not solved.

5. Use Export Variable to "flag" the variables to be used in your assembly. The Export Variable property sheet displays a list of variables associated with the model.

6. Activate the other model that contains the variable(s) you want to relate to the variable(s) in the first model. In the example, the user activates `axle_1`.



7. Export the variable(s) you want to relate to the variable(s) you exported from the other model. Use the procedure outlined in Steps 2 through 4. In this example, the user exports the variable `d1`.

Placing Component Variables in the Active Assembly

In this section, activate the root component model of your active assembly, and import the components representing the models whose variables you want to constrain in the assembly. In the example, the user is still in the `wheel-assy` assembly.

1. Select the root component in the Assembly Structure window.
2. Choose Activate Model from the pop-up menu.

   In the example, the user selects the component wheel_assy_1 and activates the model.
3. Select the List Exported and Imported Variable icon to verify the variables marked as exportable in the models. Choose Exported from the menu.
4. The property sheet appears. Type in the name of the model containing the variables you want to verify, or click on the List box and select the model from the list.
5. A Report Window appears, listing variables you exported from each model of your assembly. You can export variables from one model at a time. The Report window in this example shows the Exportable variables for `wheel` and `axle`. The output from each List Variable request is appended to the previous output in the Report window until you click Clear.

Report Window

Name	Type	Status	Value	mod-value
Exportable variables for model WHEEL				
mrad1	Length	Free	0.219000	NULL
Exportable variables for model AXLE				
d1	Length	Free	0.437500	NULL

In the report window, the Value column is the original value of the variable before it was changed. The mod-value indicates the new value after any equation in which the variable appears is changed. A `NULL` in the column indicates the value has not changed.

Symbols in the last column indicate the type of variable:

*(asterisk) - indicates the variable is associated with a parametric dimension in the model

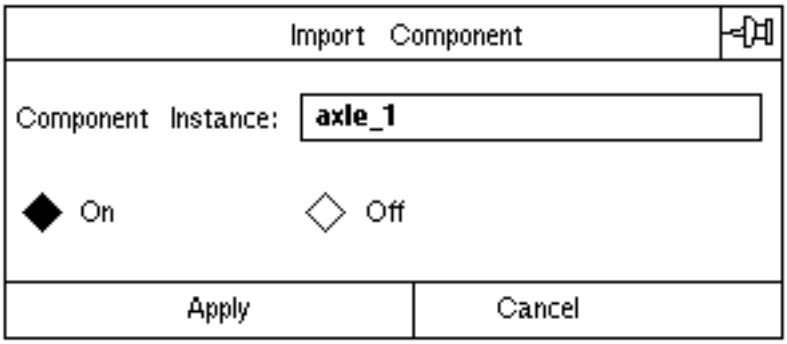
=(equals sign) indicates a static variable (a variable not associated with a parametric dimension).

Please note: If you file the assembly after you import components, any variables in the components you imported become constants. You must import them again. Before you do so, turn on a command file to save the series of functions you perform as you import variables and add equations containing the variables. You can then run the command file to export the variables.

6. Select the Import Component icon. The property sheet appears.
7. Click the pushpin so the property sheet remains open.
8. In the Component Instance field, enter the name of the component instance from whose associated model you exported variables.
9. Click Apply.

   Repeat this procedure for every model from which you exported variables.

   In the example, the user imports the `wheel_1` and `axle_1` component instances. All the exportable variables for the components specified are imported to the root component model.



Please note: If you add new variables to the model after a component is imported, you must unimport the component, then import it again to have the new variables included. If you do not do so you receive a message:
`WARNING: Component already imported.`

10. Select the List Variable icon to display a list of the imported variables.
11. Click Imported from the menu that appears.

In the Report window, the information you just requested is listed below the information from previous requests. This continues until you click Clear. Save the information from the report window by opening a report file before you select List Variable.

For example:

Report Window

Name	Type	Status	Value	mod-value	
Exportable variables for model WHEEL					
mrad1	Length	Free	0.219000	NULL	*
Exportable variables for model AXLE					
d1	Length	Free	0.437500	NULL	*
**********					
Current assembly imported variables.					
axle_ _ _d1	Length	Free	0.437500	NULL	*
wheel_ _ _mrad1	Length	Free	0.219000	NULL	*

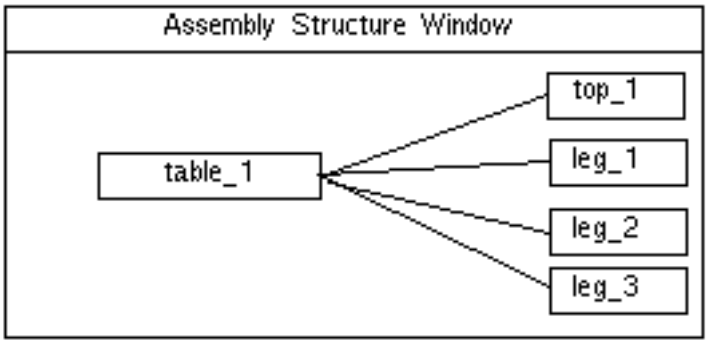
The system has assigned extended names to the variables when they were imported. The following pages discuss the rules by which these names are assigned.

Constraint Variable Names of Imported Components

The system assigns extended names to the specified variable(s) in the imported components according to the following rules:

- Do not use numbers in variable names.
- The variable name in the assembly is made up of the name of the component, three (3) underscores, and the name of the variable from the model.

  For example, for a variable whose name in the model (associated with the `top_1` component) is LENGTH, the corresponding variable in the assembly is named top___LENGTH.



- Component names can contain periods and dollar signs.

  These characters are not allowed by the constraint system. The system substitutes an underscore (_) for any of these characters when it creates the extended variable name, with one exception.

Please note: Underscores are not allowed in the first position of a constraint variable name. If a metacharacter exists in the first position of a component name, it is mapped to the letter X in the extended variable name.

- A numeric character in the first position of a component name causes errors.

  Do not use a numeric as the first character of a component. The constraint system interprets the name as a real number. To avoid this, begin component names with an alpha character.

The following table shows examples of extended variable names.

Table 9-1 Examples of Extended Variable Names as Mapped by the System

Variable Name	Component Name	Intermediate Form	Final Extended Name
length	top	top___length	top___length
length	table.top	table.top___length	table_top___length
length	table.&op	table.&op___length	table__op___length
length	table.to$	table.to$___length	table_to____length
length	=tabletop	=tabletop___length	Xtabletop___length

Preparing to Add Equations Between Variables

Before adding equations to establish relationships between variables, decide whether you want to have equations solved in automatic mode or in manual mode. This section explains what happens when you solve equations and change variable values. Subsequent sections describe the steps you must follow to add equations and change variable values.

Perform the following steps:

1. Choose whether equations are to be solved automatically (the default) or manually.
2. Add equations.
3. Solve all equations. (Do this only if you are in manual mode.)
4. Solve all components.
5. Regenerate all components.

When Are Equations Solved?

To choose either Automatic or Manual mode for solving equations, click Setup from the top bar. Click Solution. This displays the Solution Setup Property Sheet. The default mode is Automatic.

- In Automatic mode, any equation added to establish a relationship between variables is solved as soon as you click Apply on the Add Equation property sheet.
- In Manual mode, equations are solved in a separate step after you add them using the Calculator icon.

In the active assembly, you add equation(s) while active in the root component. The equation is solved (either manually or automatically) in the root component. No changes are mapped back to the respective models.

Solving Equations Manually

To solve equations manually, select the Calculator icon. A menu appears. Choose Solve Equation.

When you choose Solve All Equations, all equations in the root component of the active assembly are solved.

Changes to the Variables in the Associated Parts

Whether you solve equations manually or automatically, solve the component to have changes to variable values reflected in the model. Solve Component is a separate step. To see the changes to the model reflected in the model graphics on your screen, regenerate the component.

A variable value can change:

- As a result of solving equations
- If you change its value with Change Variable

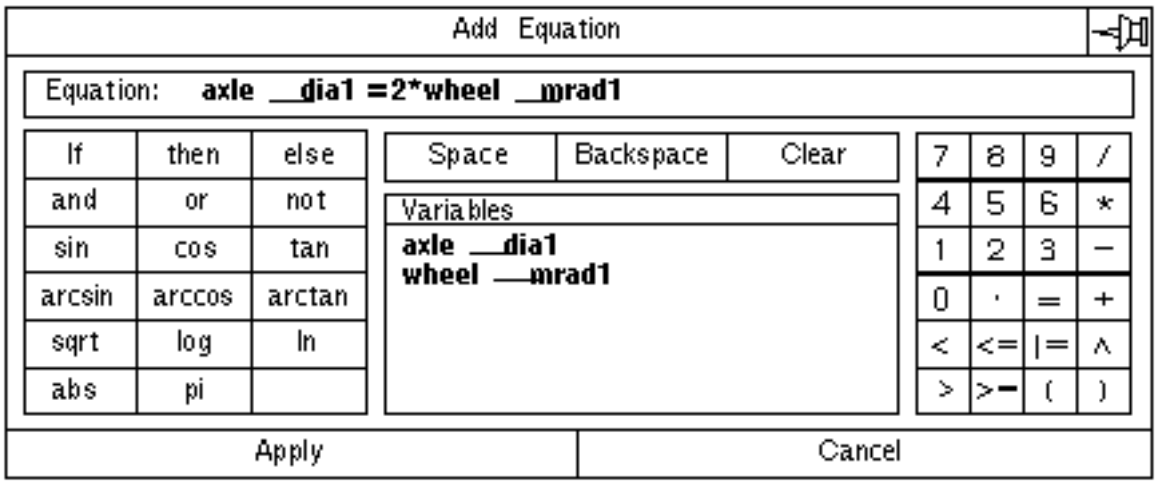
In both of the above cases, you must solve the component to have the change reflected in the model.

The section “Changing Variable Values and Applying the Changes” on page 9-17 explains these procedures in detail.

Establishing Relationships Between Variables

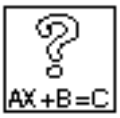
From the root component model, choose Add Equation to create equations to establish constraint relationships between imported variables.

1. Select the Add Equation icon from the Constraints task set. The Add Equation property sheet appears.
2. Enter an equation in the Equation field. For a list of variable names, click the Variables field.



Please note: In the equation, use the extended variable names that were generated by the system when you selected Import Component.

For the example, the user constrained the variable `wheel___mrad1` from the `wheel` model to the variable `axle___d1` of the `axle` model.



To see what happens as the equations are added, select the List Equation icon.

Symbols in the last column indicate the status of any equation(s) that contain the variable:

*(asterisk) — indicates the equation has been solved.

!(exclamation point) - indicates the equation has not been solved.

In Automatic Mode

If you are in automatic mode, you can see the following in the Report window.

Report Window

eq id	Text	
1	axle_ _ _d1 = 2*wheel_ _ _mrad1	**

If you select the List Variable icon, you see what happened to the variable values when the equation is solved.

Name	Type	Status	Value	mod-value	
axle_ _ _d1	Length	Free	0.437500	0.438000	*
wheel_ _ _mrad1	Length	Free	0.219000	NULL	*

In Manual Mode

If you are in manual mode, this is what you see in the Report window when you list your equations.

eq id	text	
1	axle_ _ _d1 = 2*wheel_ _ _mrad1	!

List variables. Note that no changes have been made.

Name	Type	Status	Value	mod-value	
axle_ _ _d1	Length	Free	0.437500	NULL	*
wheel_ _ _mrad1	Length	Free	0.219000	NULL	*

Solve equations, then list variables. You can see the changes.

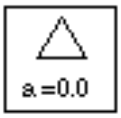
Name	Type	Status	Value	mod-value	
axle_ _ _d1	Length	Free	0.437500	0.438000	*
wheel_ _ _mrad1	Length	Free	0.219000	NULL	*

Changing Variable Values and Applying the Changes

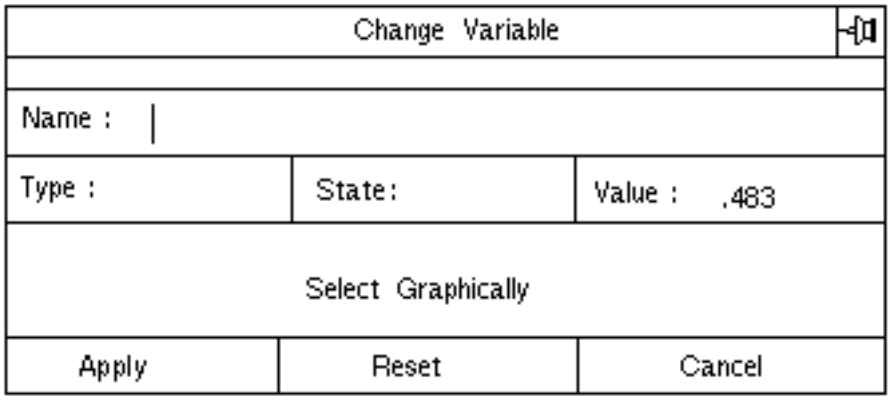
Use Change Variable to change the value of a variable of an imported component. Use Solve Component to apply the change to other variables that are constrained to the variable whose value you changed.

Changing Variable Values

Follow this procedure to change the variable values.



1. Select the Change Variable icon from the task set on the left side of the screen. A property sheet appears.

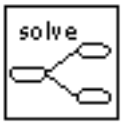


   The property sheet contains the variable name and its current value (in this example, the variable name is `axle_ _ _d1` and the value of the variable is .438).

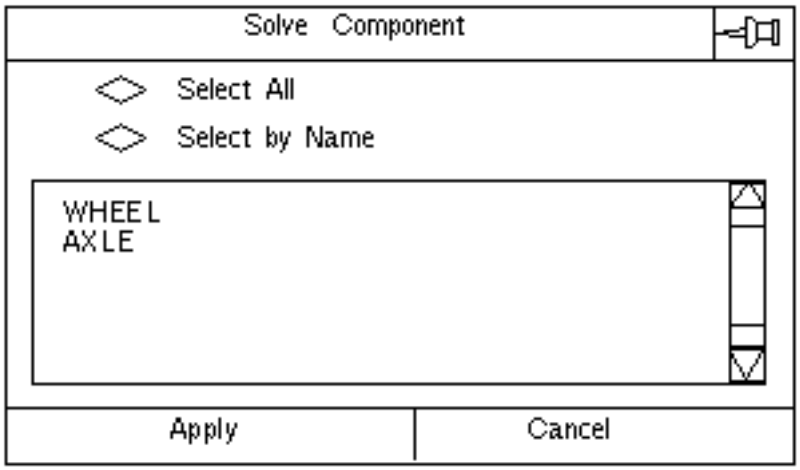
2. Change the value of the `axle___d1` variable to .5.
3. Choose the option Select Graphically to pick the parametric dimensions to perform the changes on the screen. Select Graphically works for parametric dimensions in the root node. It does not work on viewed-in parametric dimensions in the same root node.

Applying the Changes (Automatic Mode Only)

After you change the value of an imported variable, use Solve Component to apply the change to other variables that are constrained to it.



1. Select the Solve Component icon from the task set. The property sheet appears.
2. The box lists all the components that have variables that are affected by the variable value change.



Please note: If the equations are in manual mode, solve them before you select Solve Component.

3. Click Apply. The new values from the assembly are applied to all the associated models.

   If parameters are visible, the modified parameters are highlighted in red. Regenerate the component to see the changes in the representation of the model geometry in the Graphics window.

You see the following when you list variables:

Report Window

Name	Type	Status	Value	mod-value	
Exportable variables for model WHEEL					
mrad1	Length	Free	0.219000	NULL	*
Exportable variables for model AXLE					
d1	Length	Free	0.437500	NULL	*
**********					
Current assembly imported variables.					
axle_ _ _d1	Length	Free	0.437500	NULL	*
wheel_ _ _mrad1	Length	Free	0.219000	NULL	
**********					
axle_ _ _d1	Length	Free	0.437500	0.500000	*
wheel_ _ _mrad1	Length	Free	0.219000	0.250000	

In Manual Mode

The status “User changed” indicates the variable value has been changed. Solve the equations before you solve the components.

Name	Type	Status	Value	mod-value	
axle_ _ _d1	Length	Free	0.437500	0.438000	*
wheel_ _ _mrad1	Length	User-changed	0.219000	0.500000	*

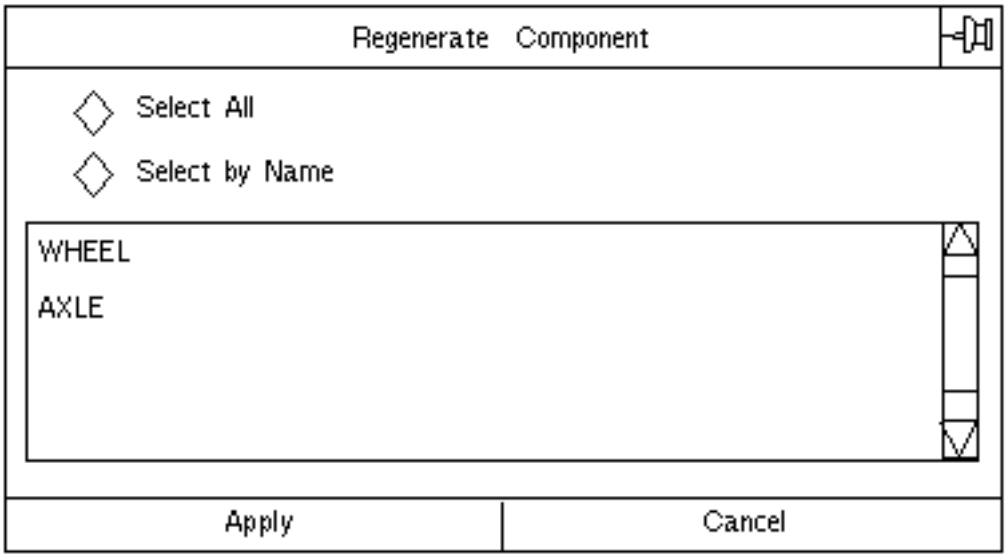
Regenerating the Component

If you have made changes to constrained variable values in your assembly and solved those components, the values for corresponding variables in your model are also changed. However, you must regenerate the component to see these changes in the representation of model geometry in the Graphics window.

You must solve a component before you regenerate it.

1. Select the Regenerate Component icon from the task set.

   The property sheet appears, listing all components for which you have imported variables.

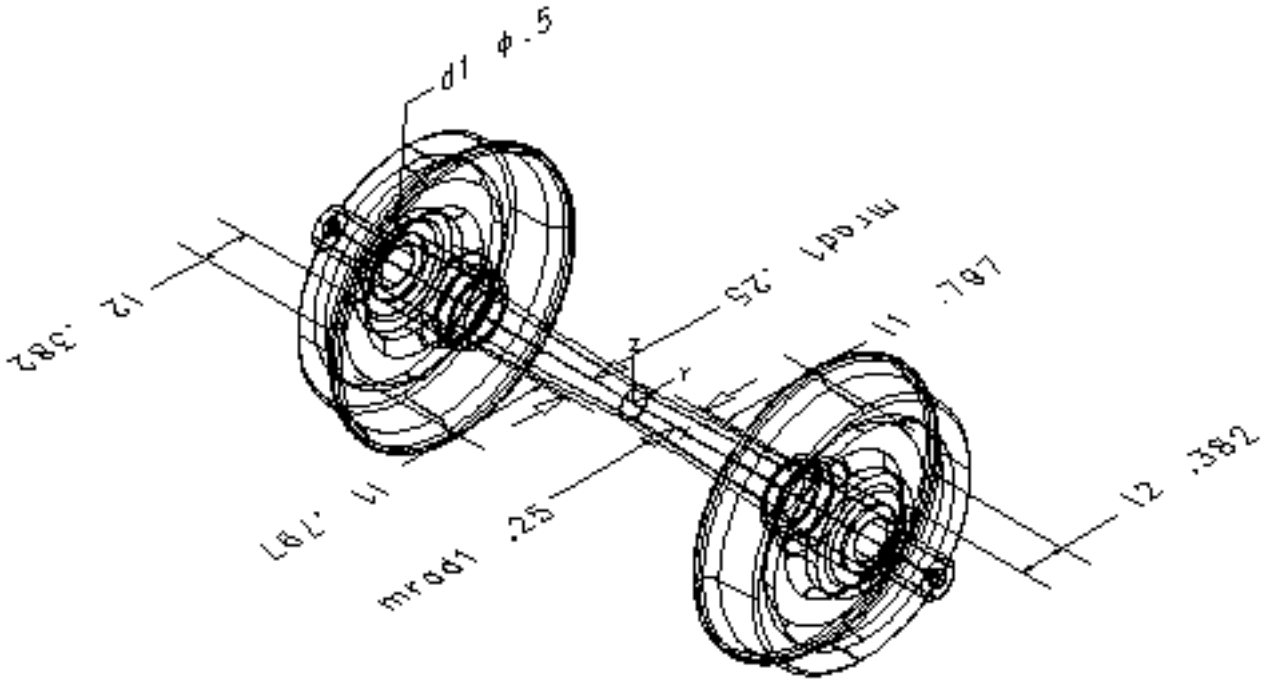


2. Select the components you want to regenerate: all components on the list, or specific components. Click Apply.

   The list can contain both solved and unsolved components. If you attempt to regenerate both solved and unsolved components, the system regenerates the solved ones, and gives you a message: `No Parameters Modified` for the ones you have not solved.

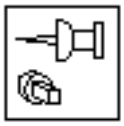
The figure that follows shows the changes to the graphic representation of the assembly.

In the example, the Graphics window shows the changes to the constrained variables.

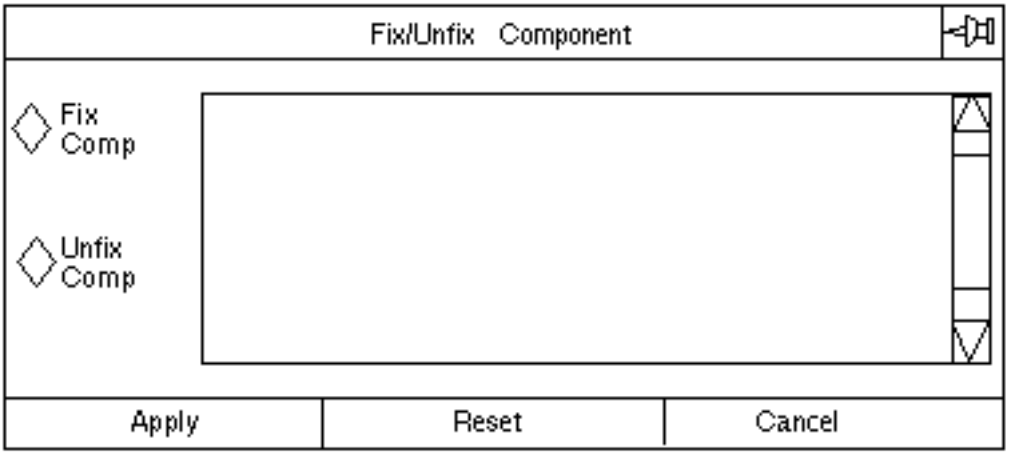


Forcing a Variable to Remain Constant

Any variable for a component can be forced to remain constant even though equations in which it is constrained to another variable have been solved.



Select the Fix Component icon from the Constraints task set.



You can directly modify the value of a variable for components that have been fixed by setting its value with the Change Variable option.

For example, you have the following equation that uses three variables: `var1`, `xcomp1_ _ _e`, and `ycomp1_ _ _h`.

var1 = (xcomp1_ _ _e + 6) * (ycomp1_ _ _f - 2)

If you change the value of `var1`, then solve the equation, the value of `xcomp1_ _ _e` also changes.

If you fix the variable `xcomp1_ _ _e`, then change the value of `var1` and solve the equation, the value of `ycomp1_ _ _h` changes. The value of `xcomp1_ _ _e` does not change, because `xcomp1_ _ _e` is now constant.

Deleting a Variable

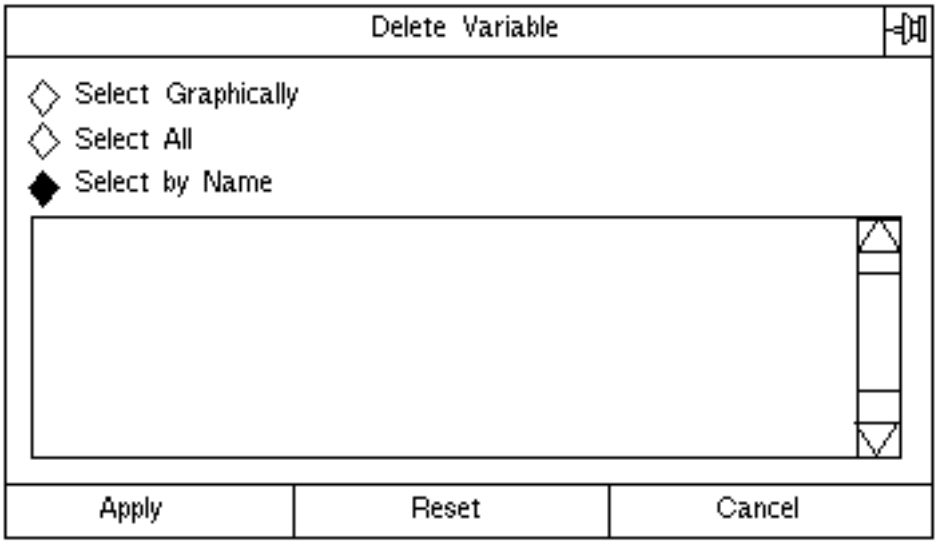
Use Delete Variable to delete a variable from the active model. Various rules apply when you delete a variable. If you delete a variable that is:

- Used in an equation, the equation is also deleted
- Used in more than one equation, all of those equations are deleted
- Associated with a parameter, it is removed from the parameter

You cannot UNDO the delete variable command.

1. Activate the model in which the variable is associated with a parameter.

2. Select the Delete Variable icon.



If you attempt to delete an imported variable, you receive an error message:

```
Variable variable-name is protected from deletion.
```

To delete a variable that is imported, use Import Component OFF to “unimport” the model associated with the variable.

Chapter 10

Adrawings and Alternate Views

This chapter describes the functions of Adrawings, how to view geometry, and how to use the Exploded View feature to show components of the assembly geometry in different locations. The topics are:

- Adrawings
- Viewing Models in Adrawings
- Using Alternate Views in Your Design
- Basic Steps to Create an Alternate View

Adrawings

When you activate a new assembly, you actually activate an Adrawing named DEFAULT. While in an assembly, you can activate additional Adrawings.

The Adrawing is a special type of CADDS part which also includes drawing information. It allows you to view the model geometry that exists in your assembly. The Adrawing keeps information about the view you select to look at the model of your assembly.

Each new Adrawing you activate gives you a “clean sheet” to define view information, such as scale, orientation, and layer assignments of the viewed assembly components.

Significance of the Active Adrawing

Each Adrawing provides:

- A way to view selected models at a particular orientation and scale, properly positioned in the assembly.
- A placeholder for newly added assembly items, such as text, notes, callouts, and reference dimensions.
- A unique alternate (exploded) view state to illustrate models at locations and/or orientations other than their assembled positions (standard view state). In the alternate (exploded) view state, you can place the models associated with component instances at any location and any orientation for the purpose of illustration and test only. You can easily toggle between the alternate view state and the standard view state.

Please note: The standard view state illustrates model instances in their actual locations and is constant for all Adrawings in the assembly. The exploded view state enables you to alter the location and orientation of each visible model instance in order to represent alternative positions - such as open and shut or an exploded assembly representation. The exploded view state is unique for each Adrawing in the assembly.

The Adrawing Provides Consistency

As you activate and access the various CADDS parts that make up the assembly, the assembly database retains information about how you look at the geometry. This includes which component instances you have viewed (VIEW COMPONENT) as well as view scale, orientation, location, and so forth. In addition, the Adrawing retains information such as layer discrimination and layer residency of all components viewed at the time you last filed. This provides a consistent environment in which to work on the assembly and its component parts.

The Default Adrawing

When you activate a new assembly, you activate an Adrawing named DEFAULT. You can specify characteristics on the Activate New Assembly and Adrawing property sheet, or take the defaults, which are displayed on the property sheet.

Adrawings and the Alternative View State

Each Adrawing in an assembly permits the display of associated models in two separate positions:

- Actual (standard view state)
- Alternate (exploded view state)

Once you have toggled the view state to exploded you can use the Move Component property sheet to define an alternate position for some or all of the associated model instances.

Views on the Adrawing are not altered with respect to scale, orientation, size, location, and quantity when view state is toggled.

Please note: The first time you toggle to Exploded for an Adrawing, you see the assembly in its standard orientation because you have not yet specified alternate positions for associated model instances.

Please note: Views on the active Adrawing are not affected when the view state is toggled between standard and alternate (exploded). The term Exploded View is somewhat misleading. In the alternate or exploded view state, it is the position of viewed model instances that can be affected - not view scale, orientation, and so forth.

Options on the Property Sheet

If you do not use the defaults on the property sheet for activating a new assembly, you can specify the following:

- The name of the assembly. Must be of 20 characters or less between directory delimiters (dots) and a maximum of 60 total characters including the absolute path name and punctuation. When you activate a new assembly, an Adrawing named `DEFAULT` is automatically created/activated.
- Assembly package toggle buttons. CAMU is already selected by default (if the CV_ASSEMBLY_PACKAGE environment variable has been set to CAMU in your `.caddsrc-local` file).
- The drawing size. Click the drawing size you want; it appears in the Form box. The default is C-1view.
- Unit of measure for the Adrawing. Click the button next to the type of units you want. The default is IN (inches).
- The name of a template Adrawing part which has been previously created.
- The design environment. The default is Assembly in Parametric. Select the ENVIRONMENT box. A menu appears for you to select your environment.

Annotating the Adrawing

You can insert CADDS entities into the active Adrawing to annotate or embellish model display. Entities that physically reside in an Adrawing are only visible while that Adrawing is active (see "Dimensioning and Annotating" on page 12-21).

Obtaining Information About Adrawings

You can list information about Adrawings in the current assembly.

1. Select Adrawing in the top bar.
2. Select List Adrawings from the resultant pulldown menu.

Sample output is shown below:

```
List adrawings for Assembly tree: Camera
Adrawing Name: =USERS.SPATIL.PARTS.CAMERA.DEFAULT
Viewing state: defined
Orientation: Normal
Status: active by prevelas
```

Viewing Models in Adrawings

When working on an assembly, you work in either an active Adrawing or an active model.

In an active model, you can see and make changes to the model's geometry. In an Adrawing, you must use View Component to view the associated model geometry.

The Adrawing contains information about the scale, model units, template, and so forth that define how you view the active assembly and its component model information. This is called the view state of your assembly. When you view components of the assembly, the Adrawing uses the information you have previously defined.

Each new Adrawing you activate gives you a "clean sheet" on which to define view information. You then view the components you want to have appear in the Adrawing.

When you activate a different Adrawing or model, the Adrawing or model in which you are just active is suspended. Later, when you activate the suspended Adrawing, it retains the view state.

Using Alternate Views in Your Design

There are two view states possible on any Adrawing. One is the standard view state. The other is the alternate or exploded view state. The standard view state allows you to view components in their actual positions - wherever you have positioned them. The exploded or alternate view state lets you “fudge” or “fake” the position of a viewed model instance. You can toggle between these two states. The standard view state is global across all Adrawings in the assembly. The toggled view state is unique to each Adrawing in the assembly (see “Adrawings and the Alternative View State” on page 10-3).

The scale, model units, location of the origin, and so forth are the same for both the view states in the same Adrawing. Views on the active Adrawing are not affected when you toggle to the alternate (exploded) state. The term “Exploded View” is somewhat misleading. In the alternate or exploded view state, it is the position viewed model instances that are affected, not view scale, orientation, and so forth.

The actual positions of the components in the assembly do not change, regardless of how you display them in the exploded view.

You can use alternate views to isolate parts, separate parts for clarity, create an “exploded” view and use them with Hide and Plot commands to produce an image. For example, you can display different operational states. In a car design, you can show the door open in the exploded view, and closed in the standard view.

In the exploded view, you can move each component as you wish to establish its position (see “Moving a Component” on page 8-24 for more information). Note the following differences between the standard and exploded view states.

- Instances of a class can have different Exploded view positions and orientations.
- Component instances of a reference assembly can be positioned and oriented in the exploded view.
- When you reposition or orient subassembly components in the exploded view, the descendants are also repositioned/oriented.

Please note: When you are moving components that have no associated CADDS models, the following message appears in the Text window

```
WARNING: The model part “PISTON-ASSY” does not exist
         You are now using the handlepart.
```

Basic Steps to Create an Alternate View

You can create a unique exploded view state for each Adrawing in an assembly (see "Using Alternate Views in Your Design" on page 10-6).

This procedure assumes that you are working in an existing assembly that contains components with associated model geometry. It tells you how to create an alternate view state. The sections that follow explain the procedure in more detail.

1. Activate an assembly.
2. Activate a new Adrawing. By default, you are in the standard view state.
3. View the components you want to include in your exploded view state.
4. Position the components in the Standard view state before toggling to the Exploded view state (see "Moving a Component" on page 8-24).
5. File the assembly and remain active in the Adrawing.
6. Click Adrawing from the top bar. From the menu that appears, click Toggle Exploded View.
7. Use selections from the Move Component icon to move and/or orient components to an alternate position.

Please note: You can toggle back and forth between the exploded and standard view states in the active Adrawing.

Please note: Use Delete Exploded View to remove the selected component instance's alternate positioning. This returns the component instance to its standard orientation.

The Standard View

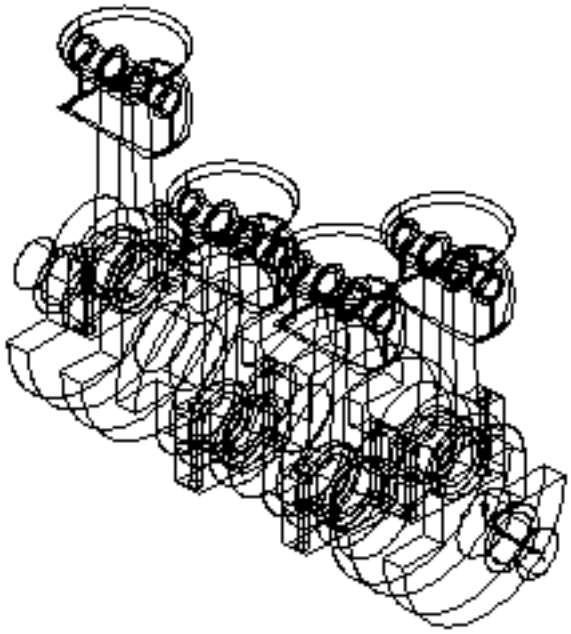
In the Standard view orientation of the active Adrawing, locate the components in their assembled positions. You can do this in several ways:

- View and position each component instance as you add new components to the assembly.
- Move components using selections from the Move Component property sheet. Some of the selections automatically view the component as it is put in its new location. Other options require that you view the component, then locate it.

The example below shows the `crank-assy` in the Standard view state. Components are properly located. To create an exploded view state, you can choose to activate a new Adrawing.

1. Click Adrawing from the top bar. From the menu that appears, choose Activate New.
2. Use View Component to view the desired components. See "Viewing a Component's Associated Model Geometry" on page 8-18.
3. Toggle to the Exploded view state. See "Selecting Exploded View" on page 10-9.

Figure 10-1 The crank_assy Assembly in Standard Orientation



Selecting Exploded View

You can toggle from the standard view state to the exploded view state: from in the active Adrawing or as you activate an Adrawing.

In an Active Adrawing

In an active Adrawing you can select a standard or an exploded view. You can use the Adrawing in which you are currently working, or you can activate a new Adrawing.

1. In the standard view, position the components in their assembled locations in the assembly (see "Moving a Component" on page 8-24).
2. File the assembly and remain active in the Adrawing.
3. Click Adrawing from the top bar. From the menu, choose Toggle Exploded. The following message appears, indicating that the toggle has just occurred:

   ```
   The current Adrawing is in Exploded orientation.
   ```

See "The Exploded View" on page 10-10.

As You Activate an Adrawing

You can activate an existing Adrawing in the Exploded view state using an option on the property sheet.

1. Open the Activate Old Adrawing or Activate Old Assembly property sheet.
2. Select the Exploded View On button. When the assembly is activated, you are in the exploded view.

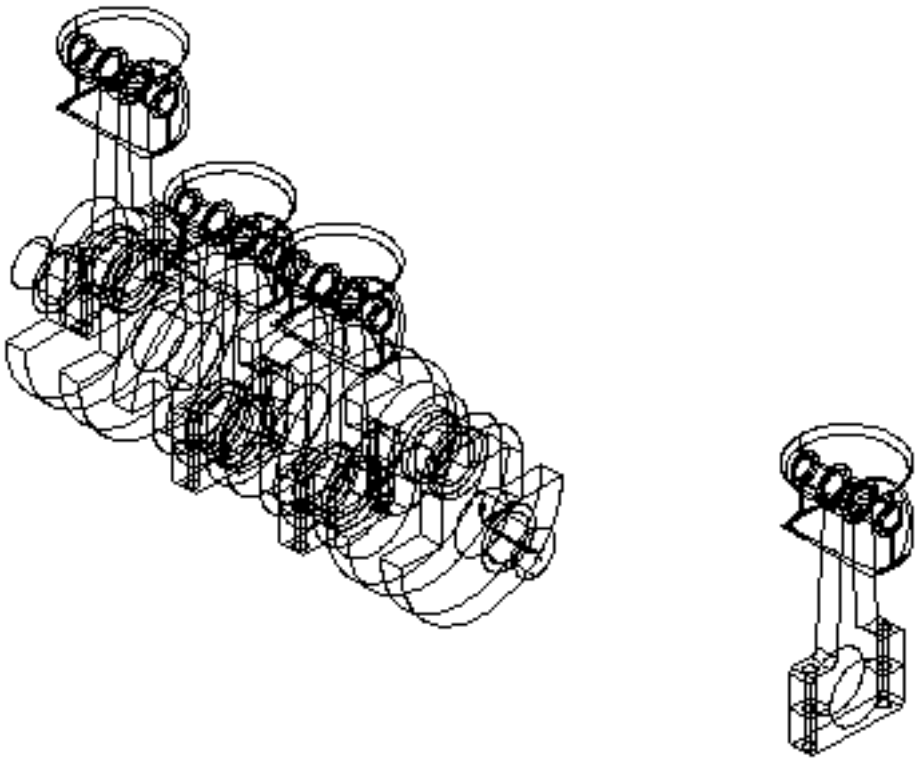
The Exploded View

When you are in the Exploded view state, you do not change the actual positions of components in the assembly geometry. You change only the appearance of their locations in the exploded view. You need not lock the component and its associated model while working in an Exploded view.

Use options from the Move Component property sheet (see “Moving a Component” on page 8-24) to reposition components in the exploded view state. To move parts in the exploded view, do the following:

1. Select the component to be moved from the Assembly Structure window. In the example, the user selects `piston-assy_1`.
2. Open the Move Component property sheet. Select an option from it. In this example, the user selects the Translate icon.
3. Relocate the instance. In the example, the user moves the `piston-assy_1` component 15 inches along the x- axis.

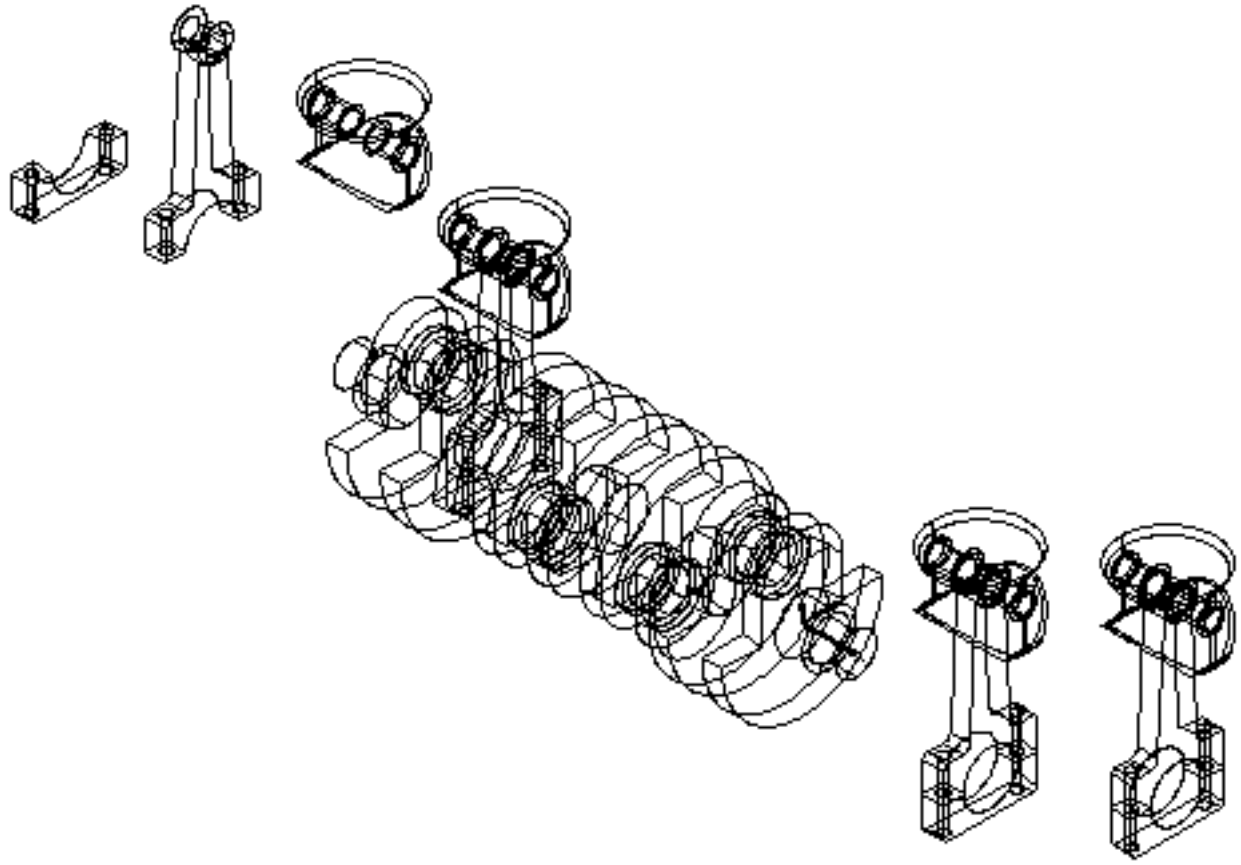
Figure 10-2 Piston-assy_1 Component of the crank_assy Assembly



4. Repeat steps 1-3 for any other desired instances.

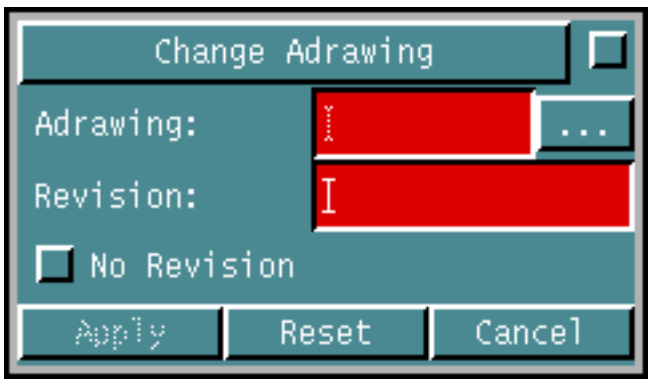
The figure below shows the completed exploded view.

Figure 10-3 The crank_assy Assembly in Completed Exploded View



Changing an Adrawing

You can change the definition of an Adrawing using the Change Adrawing option. Select the Change Adrawing option from the Adrawing menu of the top bar. The Change Adrawing property sheet appears.



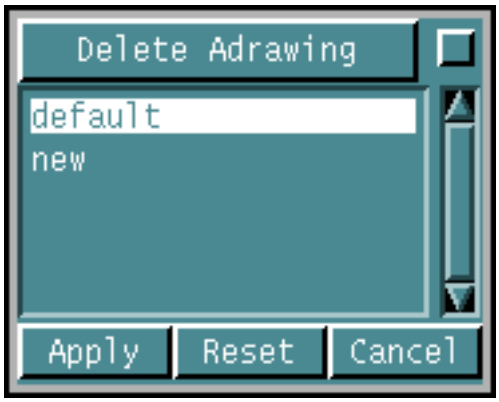
Specify the following:

- Adrawing: Specify the Adrawing name from the runtime list of existing Adrawings that you want to change.
- Revision: Specify the revision of the Adrawing part. You can use a maximum of 20 characters to specify the revision.
- No Revision: Check the No Revision option to remove the revision associated with the selected Adrawing.

Please note: You can specify the revision only if the Adrawing is not active or suspended.

Deleting an Adrawing

You can delete an Adrawing definition from the assembly database using the Delete Adrawing option. Select the Delete Adrawing option form the Adrawing menu of the top bar. The Delete Adrawing property sheet appears.



Select the desired Adrawing name from the runtime list to delete the Adrawing. Click Apply to confirm deletion of the adrawing.

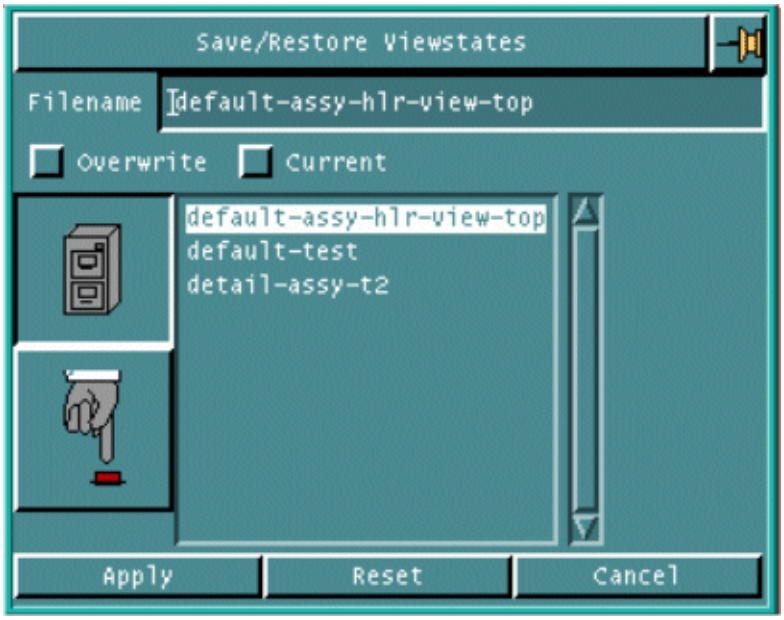
Saving and Restoring Viewstates

You can save the viewstates of the components of an assembly either to the active part or the Default Adrawing directory. You can also restore the viewstate information from a saved file.

The viewstate file stores both the component instance name and the CAMU-id to enable EPD.Connect to also work with viewstates. An error message is displayed when the component instance name is not found.

To save or restore viewstates, choose the Save/Restore Viewstates option on the Adrawing menu. This option opens the Save/Restore Viewstates property sheet.

Figure 10-4 Save/Restore Viewstates Property Sheet



1. In the Filename field, specify the name of the file in which you want to save the view state information or from which you want to restore the information.

   The default value is the name of the current Adrawing, followed by a hyphen. You can either retain this default value as part of the file name, or specify another file name.

2. Select Current to save the file in the `/<active-part-dir>/_bcd` directory of the active part.

   If no part is active, then by default the file is saved in the `_bcd` directory of the active Adrawing of the active assembly.

   If a part is active and you do not select Current, the file is saved in the `_bcd` directory of the Default Adrawing of the active assembly.

3. Alternatively, select Overwrite if a file with the specified name exists, and you want to overwrite it with a new one.

4. Select if you want to save the viewstate information into the file.

5. Select if you want to restore the viewstate information from the selected file. You can select from the list of files, where the viewstates of the current assembly are already saved in the Default Adrawing directory or the directory of the current part. You can also enter the name of the file in the Filename field.

   These viewstate files are independent of the Adrawing. You can also restore the viewstate of one Adrawing to another Adrawing.

Please note: In case a part has viewstate files from multiple assemblies, you must select the relevant viewstate file.

Chapter 11

Defining Nongraphic Information for the Assembly

This chapter describes how to define attributes, which are data you define and associate to components of the assembly structure. You can extract this information with CVMAC programs. The topics are:

- About Attributes
- Creating Attribute Groups
- Associating an Attribute Group with a Component
- Disassociating an Attribute Group from a Component
- Listing Attribute Groups
- Creating Attributes
- Editing Attributes
- Locking Attributes and Attribute Groups
- Changing Attribute Properties
- Removing an Attribute
- Removing an Attribute Group
- Generating Output Using Supplied CVMACs

About Attributes

Attributes are user-defined data you assign to one or several components of the assembly structure. For example, you can assign bill-of-materials information, physical properties, drawing notes, or labor tracking data. To produce reports with this data, use a CVMAC program you have written, or use one of the programs supplied with CAMU (see “Supplied CVMAC Programs” on page 11-19).

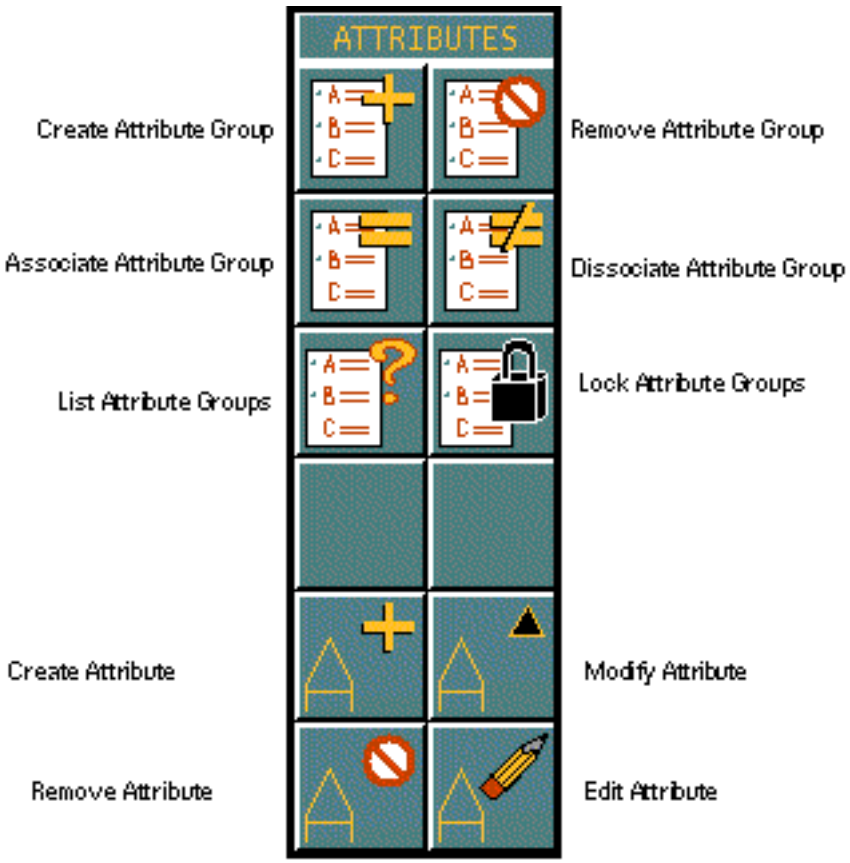
You can use the Attribute feature only from the Explicit Assembly environment.

Before you begin creating and defining attributes, think about the kinds of reports or lists you want to generate with the attribute information you assign to your assembly. For example, you might wish to produce a bill of materials.

Attributes Task Set

The Attributes task set is available in the Explicit Assembly environment.

Figure 11-1 Attributes task set options



Overview of Procedure to Use Attributes

Define an attribute group, which can be the name or category of one of the reports you want to produce with the data assigned to components. For example, choose a group name BOM to represent the bill of materials you want to produce.

Create attributes within this attribute group. Each attribute name refers to one type of information you want to store with each individual component. For the attribute group BOM, you might have the attributes part number, vendor, and description.

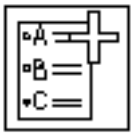
For each attribute you create, specify its name, the group it belongs to, the attribute type (integer, real number, string, or array), and whether the name is to be associated with one or all instances of the component.

Specify which components in your structure are to be associated with the group you defined. This lets you store information with the component for each attribute you named.

Creating Attribute Groups

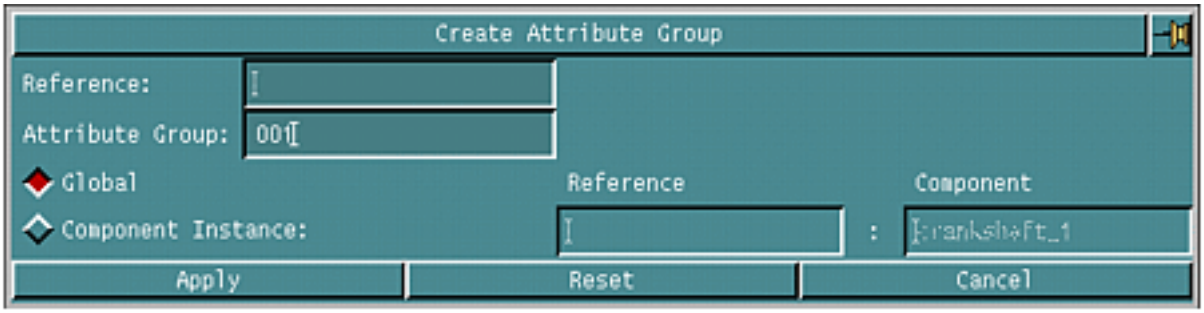
An attribute group is a category that can contain several different attributes. Create Attribute Group creates the attribute group and associates it with all components or one component instance.

When you define a group, associate it with all the components in the structure (a global group) or to one component instance. To associate with one component, the component must be locked.



1. Select the Create Attribute Group icon from the task set. The Create Attribute Group property sheet appears.

Figure 11-2 Create Attribute Group property sheet



2. To define several attribute groups in succession, click the pushpin icon to keep the property sheet displayed. Define one attribute group. Click Apply.

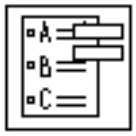
   The property sheet is still displayed for you to define the next attribute group.

3. When the cursor appears in the Attribute Group field, enter a name.
4. To have the attribute group apply to every component in your assembly structure, click the Global button.
5. To have the attribute group apply only to one instance of a component, click the Component Instance button. In the Component Instance field, type the name of a component instance with which the attribute group should be associated.
6. Fill the property sheet. Click Apply.
7. If you are defining more than one attribute group at a time, repeat Steps 3, 4, and 5.

Associating an Attribute Group with a Component

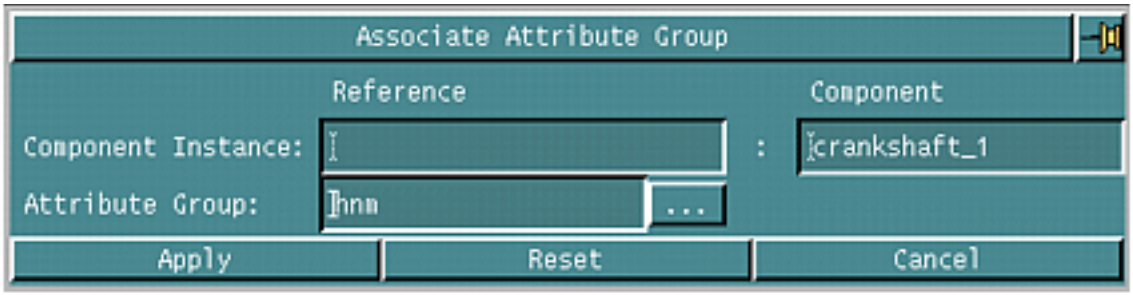
When you create an attribute group (if it is not a global group), specify the component instance with which you want the attribute group to be associated.

You can associate the attribute group to an additional component(s). The component you are associating the group with must be locked.



1. Select the Associate Attribute icon to associate an attribute group with a component instance. The property sheet appears.

Figure 11-3 Associate Attribute Group property sheet



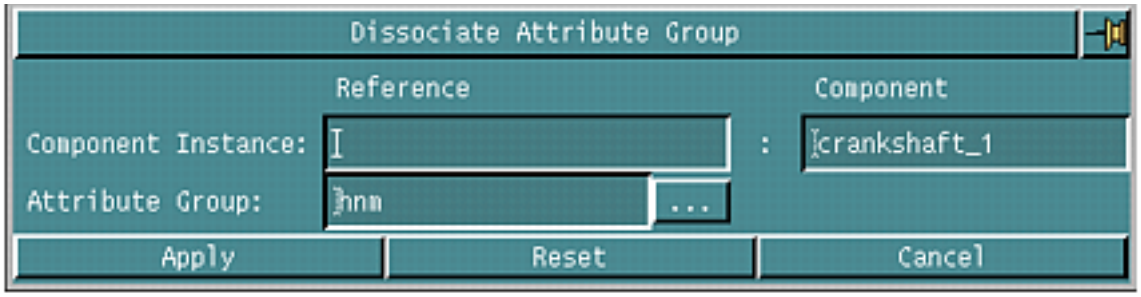
2. Specify the Attribute Group name and the Component Instance you are associating in the respective fields.
3. Click Apply.

Disassociating an Attribute Group from a Component

To remove an established association between an attribute group and a specific component, use Disassociate Attribute Group.

1. Select the Disassociate Attribute icon. The property sheet appears.

Figure 11-4 Dissociate Attribute Group property sheet



2. Specify the Attribute Group name and the Component Instance from which you are removing the association in the respective fields.

Please note: If attribute values have been assigned to a component(s), you cannot remove the association between the component and the attribute group. You must first remove the attribute value.

Listing Attribute Groups

To list the attribute groups associated with your assembly, do the following:

1. Select the List Attribute Group icon from the task set. The property sheet appears.

Figure 11-5 List Attribute Groups property sheet

2. Enter the name of the active assembly in the Assembly Name field.
3. Click All Attribute Groups, all Global groups, or all Nonglobal groups as required.

   A list of the respective groups is displayed when it is selected.

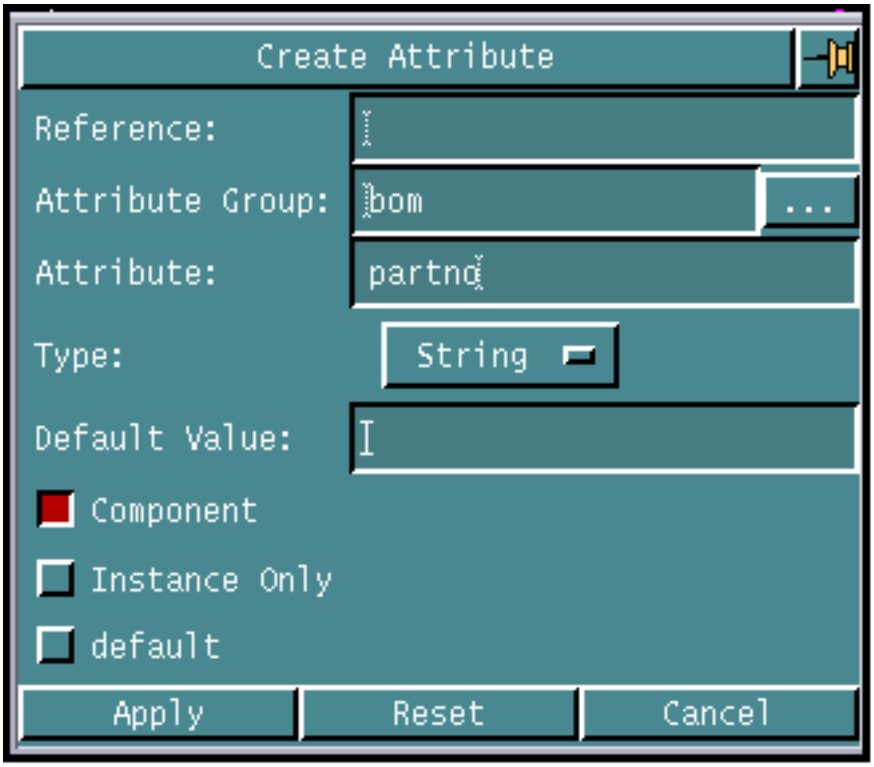
Creating Attributes

Define an attribute group. Create the attributes that are to be included in the group.

Perform the following steps to create an attribute.

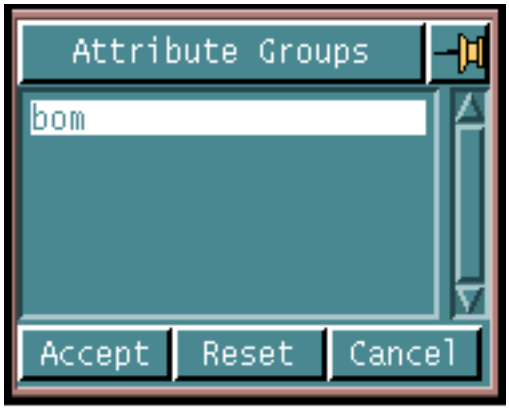
1. Select the Create Attribute icon. The property sheet appears.
2. Use the pushpin to create several attributes in succession.

Figure 11-6 Create Attribute property sheet

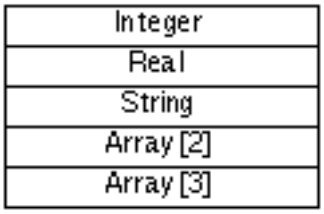


3. Type a name for an attribute in the Attribute field.

4. Click the options list button to the right of the Attribute Group field. A menu appears, showing a list of all attribute groups you have created.



5. Click the name of the attribute group you want. It appears in the Attribute Group field.
6. Click the list button next to the Type field. Choose the required attribute type.



The choices allow you to enter the following types of data:

Option	Type of Data
Integer	A number value.
Real	A double-precision floating point.
String	Letters, integer values, or any valid character. If you use a space to separate a string value, the characters must be in quotes.
Array [2]	An array of two real numbers.
Array [3]	An array of three real numbers.

Use standard decimal notation format for arrays separated by either a comma or a space.

7. Specify the default attribute value in the Default Value field. If no other value is assigned to the attribute, this default value is used.
8. Choose Component if you want all instances of the class of components to have the same value for the attribute.

9. Choose Instance Only if you want different instances of a component class to have different values for this attribute.
10. Choose Default if the value in the Default Value field is to be used. The attribute has the same value for all component classes that have this attribute associated with them.

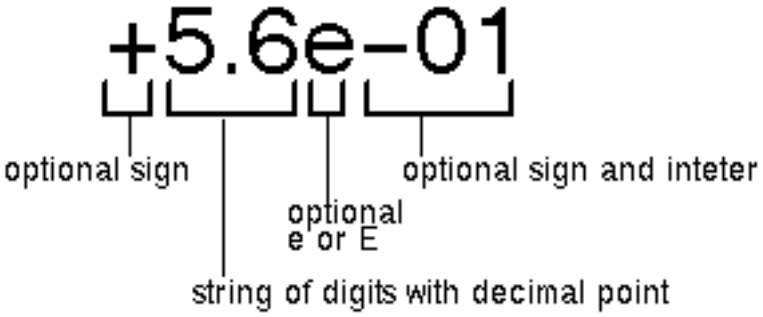
Floating-Point Numbers

The input format for arrays can also be a string of floating-point numbers separated by either a comma or a space. For example, for Array(2) you could have -3.5,+5.6e-01 or -3.5, 5.6E+01.

See the example below.

The floating-point number format may contain the following:

- An optional sign
- A string of digits, optionally containing a decimal point
- An optional e or E
- An optional sign and an integer



If the number of input values is less than the number of values in the array, the remaining elements have the value 0.0. For example, if the user inputs the following default values: -3.5,5.6E+01 for an Array (3) data type, the first element has the default value of -3.5, the second has the value of 56 and the third has a value of 0.0.

Editing Attributes

Edit attributes to assign values to them.

1. Select the Edit Attribute icon. The property sheet appears.

Figure 11-7 Edit Attributes property sheet

2. Specify the name of the component instance whose associated attribute(s) you want to edit in the Component Instance field.

   You can specify a reference Assembly Instance name for a read-only copy of the attribute editor.

3. Click the Component button to edit the entire component class to which the component instance belongs.
4. Click the Instance Only button to edit only the component instance you specified.
5. Click Apply. The Attribute Editor window interface appears.

Attribute Editor Window

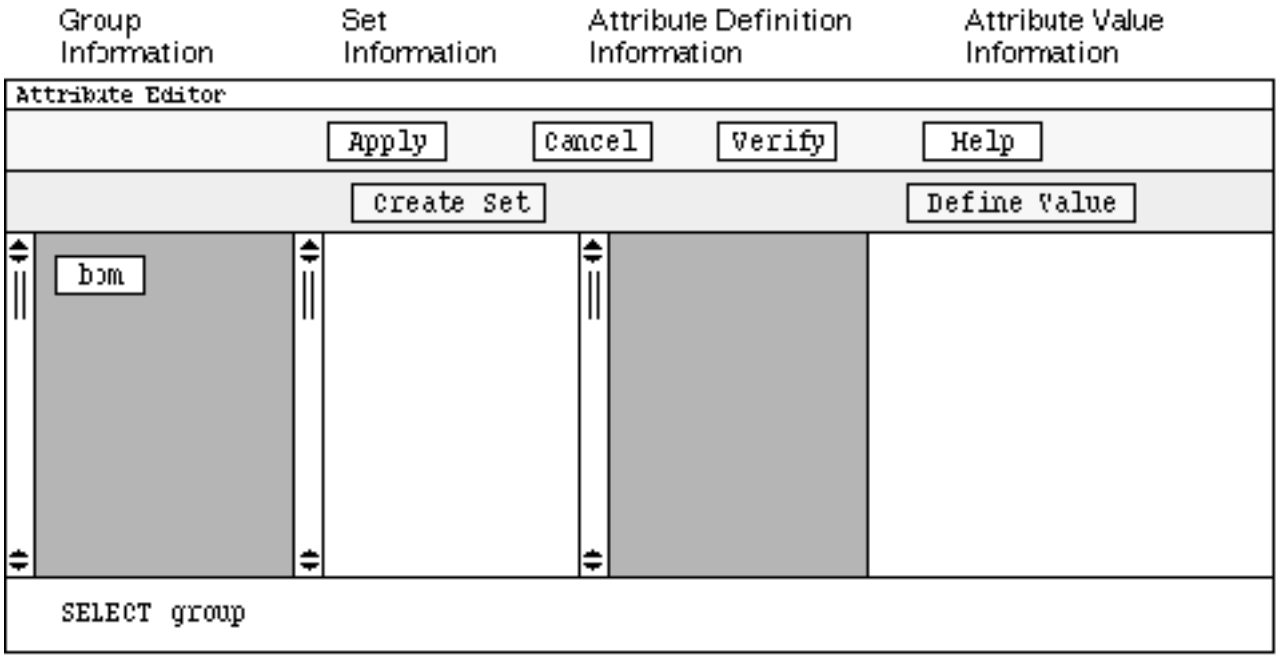
Assign values to attributes using the attribute editor window interface. The top area of the Attribute Editor contains four pushbuttons.

- Apply - Applies changes made with the Attribute Editor during the current session. Changes are saved in the active assembly structure.
- Cancel - Cancels all changes made with the Attribute Editor during the current session. Changes are not saved in the active assembly structure.
- Verify - Displays the edited object's attribute information in the Text window.
- Help - Displays a help file for using the Attribute Editor.

The second strip from the top of the editor contains the selections Create Set and Define Value. These selections appear hazed when the editor first appears.

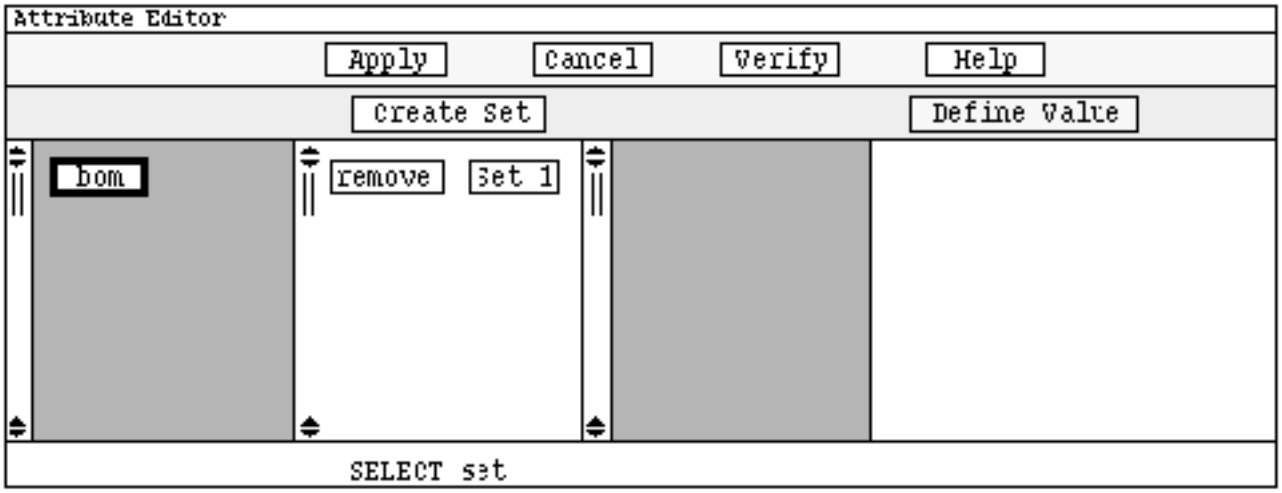
Below the second strip, the editor displays four columns of information for the component instance or assembly instance you specified on the Edit Attributes property sheet.

The bottom strip of the Attribute Editor displays prompts for you to follow as you edit attributes.



1. Select an attribute group name from the group column.

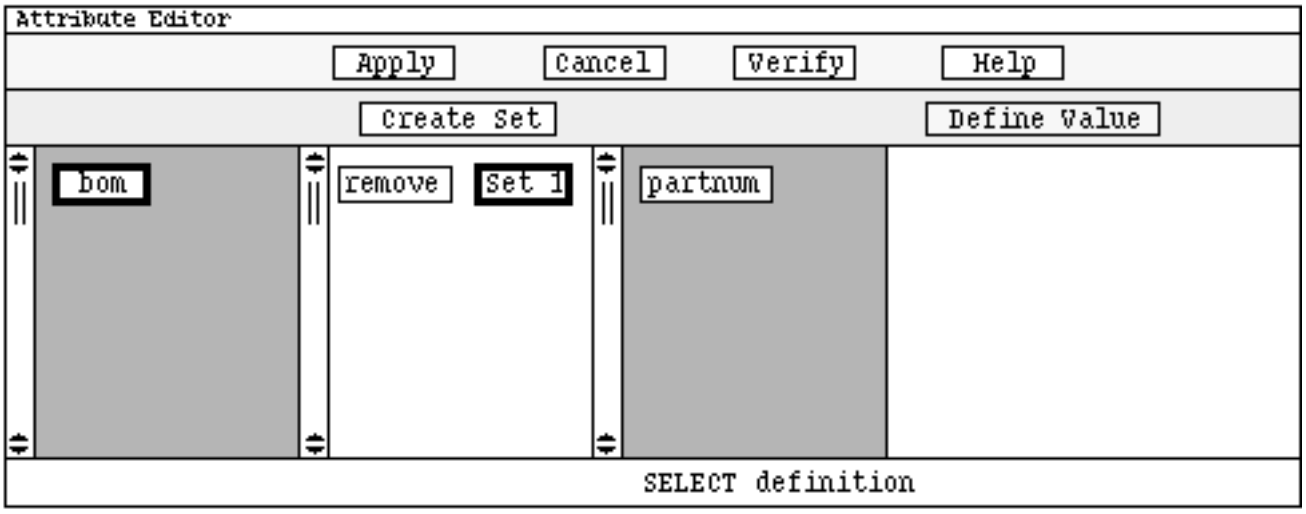
   The selected attribute group appears highlighted. The Create Set button is unhazed. Set information appears in the second column. A new prompt appears: `SELECT set`



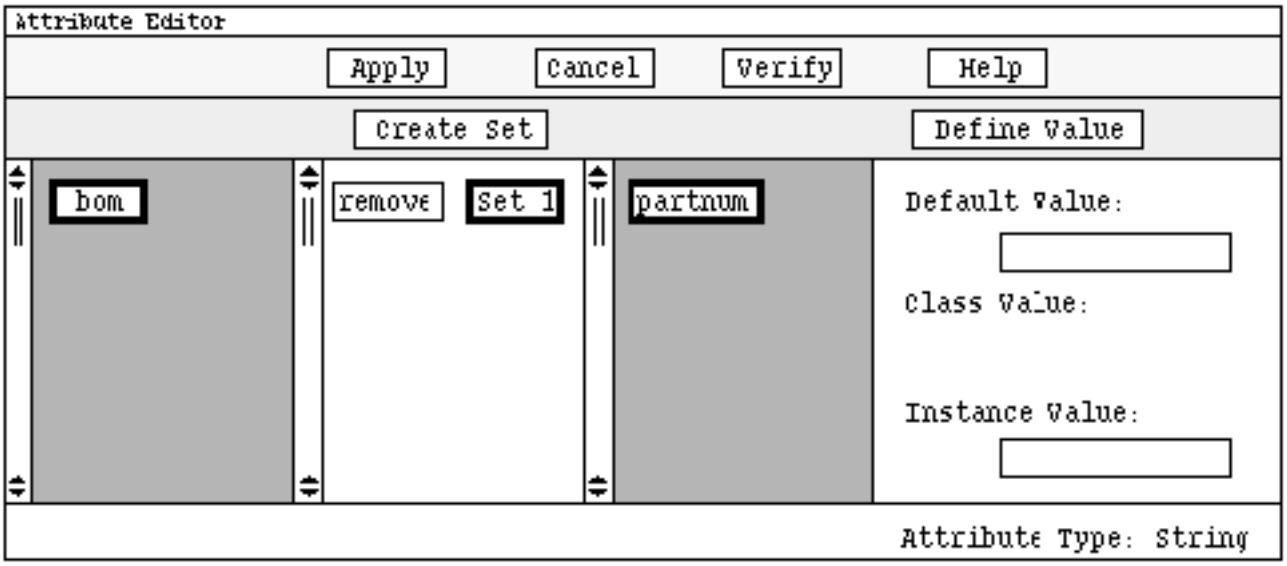
   Use different sets to assign several different attribute values to the same attribute name(s) in a group. When you select a group, set 1 is automatically created. To create additional sets, select the Create Set button. To remove a set, select the Remove button next to the set number you want to remove.

2. Select the set in which you want to add information. The button is highlighted. The attribute(s) created in the attribute group (that you selected in Step 2) appear in the third column.

   The prompt changes to `SELECT definition`.



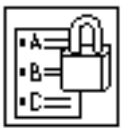
3. Select an attribute name from the definition column. The attribute selected is highlighted. The Define Value button becomes unhazed, and fields to define values appear in the fourth column. The prompt tells you the type for the attribute you are editing.



4. Click the Define Value button. Type the information. You must press Return after entering a value.
5. Select another attribute definition from the third column and assign a value to it. Press Return after entering a value.
6. Click Apply to file the assigned value information.

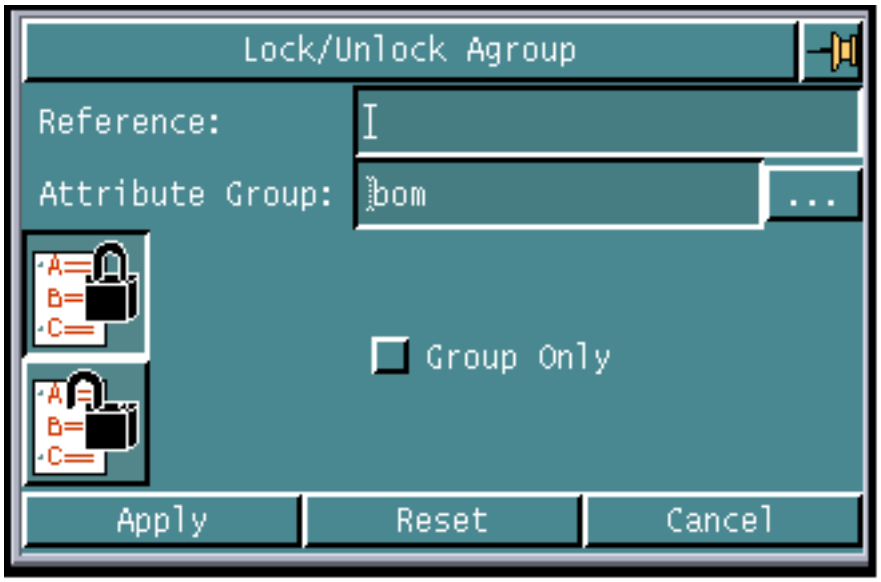
Please note: Apply the Attribute Editor property sheet when you have finished assigning attribute values for the selected component, and before you select another component. If you attempt to select another component before you apply the property sheet, you may be unable to continue your CADDS session.

Locking Attributes and Attribute Groups



To change attribute properties, or to edit attributes, you must first lock the attribute. Select the Lock Attribute icon.

Figure 11-8 Lock/Unlock Agroup property sheet



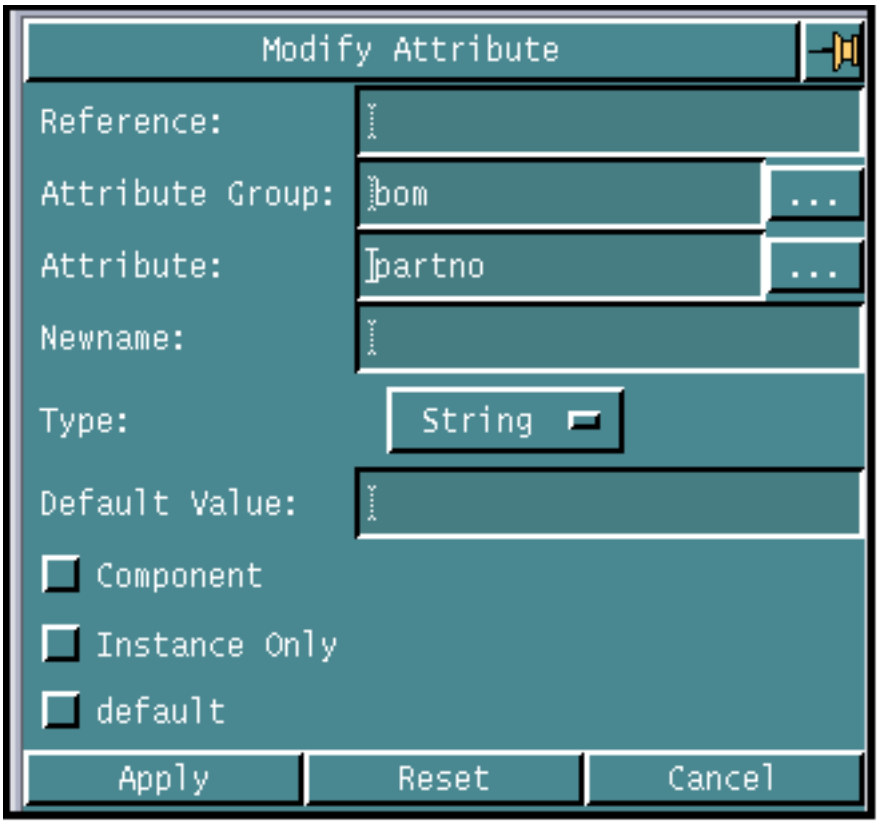
Changing Attribute Properties

You can change the properties of an attribute you have created. You must lock the attribute.

To change attribute properties, do the following:

1. Select the Modify Attribute icon. The property sheet appears.

Figure 11-9 Modify Attribute property sheet



2. In the Attribute field, enter the name of the attribute whose properties you want to change.
3. You can change the following:
   - Attribute name
   - The group with which the attribute is associated
   - Attribute type
   - Value of the default
   - Association of the attribute with one or all instances of a component class

Removing an Attribute

Use Remove Attribute to delete an attribute and all its definitions in the group. You must first lock the attribute and all its associated definitions.

1. Select the Remove Attribute icon. The property sheet appears.

Figure 11-10 Remove Attribute property sheet

2. Specify the group name to be removed in the Attribute Group field.
3. Specify the Attribute.
4. Click Apply.

Please note: If a value is associated with one of the group definitions, the system identifies the associated group and displays a warning message. You cannot remove a group that is associated with a component.

Removing an Attribute Group

To remove an attribute group and all its definitions in the group, use Remove Attribute Group. The attribute group and all its associated definitions must be locked.

1. Select the Remove Attribute Group icon. A property sheet appears.

Figure 11-11 Remove Attribute Group property sheet

2. The property sheet lists the attribute group names in the active assembly.
3. Select the group(s) you want to remove.

Generating Output Using Supplied CVMACs

To extract attribute information and produce reports, use a CVMAC program. CAMU provides CVMAC programs as examples, which you can use, copy, and modify to your specific needs. You can also write additional programs.

Supplied CVMAC Programs

These are several CVMAC programs supplied with CAMU. They are located in the `/usr/apl/cadds/data/am/supplements di`rectory.

Query

The `query` CVMAC searches and identifies all components that contain specified attribute or value. Define the assembly structure (tree), the attribute definition(s) and (optionally) value(s) you want to locate. The program prints a component list.

Files for the query program are in the subdirectory `query`.

Bill of Material

The `BOM` CVMAC prints attribute definitions and their assigned values, displaying them in column format. Yu must provide the following information:

- Assembly structure (tree) name.
- An output file name.
- A user name to be printed on the header.

Files for the attribute definitions program are in the subdirectory `BOM`.

Generate Assembly Structure Entities

The `tree` CVMAC creates a graphical assembly structure in the active CADDS part to mimic the assembly structure. It creates CADDS model mode, wireframe entities in the active part (normally an Adrawing). To run this program you must be in a part. Using the program, choose the layers on which to place the relation lines, component text, and component boxes. You may choose a rectangular or a chamfered box. This program is specifically designed to enable you to plot the assembly structure.

Files for the program to plot the assembly structure program are in the subdirectory `tree`.

Output All Attribute Information

The `amattrs` CVMAC outputs all attribute information associated with a selected object. The object can be a component, component instance, relation, or relation instance. Identify the object using its ID or its name and object type. In the output, attributes beginning with a pound sign (#) are internal attributes.

Creating a CVMAC Executable

Use this procedure to create a CVMAC executable. Examples for each step are shown. When an absolute path name is not specified, CVMAC assumes the path of the create directory. For further information about any CVMAC commands, see the *CVMAC Language Reference Manual*.

1. Compile the files found in the appropriate subdirectory using the CVMAC command CVMCOMP. For example:

   ```
   cvmcomp getvalue
   ```

2. Make a link file to create the executable program. A sample link file for the query program is shown below.

```
IMAGE =users.xxx.cvmac.query
MAIN =usr.apl.cadds.data.am.supplements.query
LINKM =usr.apl.cadds.data.am.supplements.query.addvalue
LINKM =usr.apl.cadds.data.am.supplements.query.askuser
LINKM =usr.apl.cadds.data.am.supplements.query.compare
LINKM =usr.apl.cadds.data.am.supplements.query.display
LINKM =usr.apl.cadds.data.am.supplements.query.gettree
LINKM =usr.apl.cadds.data.am.supplements.query.getvalue
LINKM =usr.apl.cadds.data.am.supplements.query.has-attr
LINKM =usr.apl.cadds.data.am.supplements.query.prntpage
LINKM =usr.apl.cadds.data.am.supplements.query.convert
LINKM =usr.apl.cadds.data.am.supplements.query.header
LINKM =usr.apl.cadds.data.am.supplements.query.findcomp
LINKM =usr.apl.cadds.data.am.supplements.query.getattr
ENDLINK
```

3. Link the files with the link file to make the executable using the CVMAC command CVMLINK. The executable is the file specified by the IMAGE statement in the link file. For example,

   ```
   cvmlink =users.xxx.cvmac.linkfile
   ```

4. In CADDS, issue the command `RUN CVMAC <filename>` where `filename` includes either an absolute (CADDS) path name or a path name below the create directory.

   The input and output of the programs use the create directory unless you give an absolute path name.

5. To modify any attribute(s) used for the BOM, change the DATA statement(s) in the BOM file `compinfo`.

Link and Compile CVMAC Procedures

To use the supplemental CVMAC procedures, they must be linked and compiled. Change to the directory `cvmac` and issue the command `make`.

Please note: You must be able to issue the command `cvmcomp` and have write access to the current directory.

Creating CVMAC Executables

This section contains the procedure (and examples) to convert supplied CVMAC attribute/report extraction programs to executable, usable files.

Procedure

Follow this procedure.

1. Copy the contents of `/usr/apl/cadds/data/am/supplements` into a subdirectory of your CADDS create directory. Name the subdirectory `cvmac`.

   For example:

```
pwd <cr>
   /usr2.pam (This is the $HOME directory)
mkdir parts/cvmac <cd>
cd parts/cvmac <cd>
cp -r /usr/apl/cadds/data/am/supplements .<cd>
ls
   README            ad-edm         BOM
   one-file-progs    query          tree
   README_1          ad-handlepart
more .caddsrc-local <cr>
  .
  .
  .

setenv CVPATH $HOME/parts=C:
```

   In this example, `parts` is the “create” (`=C:`) directory.

2. Before you run the `cvmcomp` compiler command, be aware that all of the output (compiled) files are placed in the CADDS create directory. If you want to maintain the CVMAC files in a directory other than your parts directory, temporarily change the CADDS create directory to the `cvmac` subdirectory. Edit the CVPATH in your local `.caddsrc` file, as shown in the following example.

   For example:

```
setenv CVPATH $HOME/parts/cvmac=C:
```

3. Change to the cvmac directory, list the directory contents, and issue the `cvmcomp` command for every file contained in the `query` subdirectory, as shown in the following example:

```
cd /usr2/pam/parts/cvmac/query
ls
 addvalue convert findcomp gettree has-attr
 prntpage askuser compare display getattr
     .
     .
     .
cvmcomp addvalue
cvmcomp askuser
     .
     .
     .
```

4. Issue the `cvmcomp` command for every file in both the `BOM` subdirectory and the `tree` subdirectory of the cvmac directory, in the same way you did for the files in the `query` subdirectory.

5. Make a link file for the `query` routine. Create a text file as shown in the following example:

```
vi linkquery <cr>

IMAGE =usr2.pam.parts.cvmac.doquery
MAIN =usr2.pam.parts.cvmac.addvalue
LINKM =usr2.pam.parts.cvmac.askuser
LINKM =usr2.pam.parts.cvmac.compare
LINKM =usr2.pam.parts.cvmac.display
LINKM =usr2.pam.parts.cvmac.gettree
LINKM =usr2.pam.parts.cvmac.getvalue
LINKM =usr2.pam.parts.cvmac.has-attr
LINKM =usr2.pam.parts.cvmac.prntpage
LINKM =usr2.pam.parts.cvmac.convert
LINKM =usr2.pam.parts.cvmac.header
LINKM =usr2.pam.parts.cvmac.findcomp
LINKM =usr2.pam.parts.cvmac.getattr
ENDLINK
:wq
```

6. Link the `linkquery` file with the compiled files to make the executable `query` routine as shown in the example below:

```
pwd
/usr2/pam/parts/cvmac
cvmlink linkquery <cr>
```

7. Repeat Step 5 to write a link file for both the `BOM` and the `tree` routines. Use the file name `BOM` in place of the file name `query`, and again for the file name `tree`.

8. Repeat Step 6 to link the `linkBOM` and the `linktree` files with their compiled files to make the executable `BOM` and `tree` routines.

9. The executable files `query`, `BOM`, and `tree` are now ready to use. Be sure to change your CADDS create directory to its original state. In this example, the create directory would be reset to `/usr2/pam/parts`.

10. After assigning attributes, file the assembly and perform the data extraction routines while you are active in the assembly.

11. Based on the example shown in this section, type the following CADDS Explicit commands to produce three report files:

    To run the `query` extraction routine, type:

    `run cvmac cvmac.doquery` and press Return.

    To run the `BOM` and `tree` routines, enter the same command, and replace the name `query` with the name `BOM` and `tree`.

    All files output from in the Explicit CADDS environment are placed in the `_bcd` subdirectory of the CADDS create directory. In this example, the report files would be placed in `/usr2/pam/parts/_bcd`.

12. List your cvmac directory, which is `/usr2/pam/parts/cvmac` in this example, and compare the listing with the directory listing produced in Step 1. Note the `.cm` and `.dm` files.

Chapter 12

Documenting the Assembly

This chapter describes how to generate HLR, dimensions, text, labels, and other annotations to document the assembly. The topics are:

- Creating an Assembly Drawing Document
- Generating HLR Using HIDE OBJECT
- Generating HLR Using CADDShade II
- Dimensioning and Annotating
- Plotting

Please note: For more information, see the following online books:

- *Design and Drafting User Guide and Menu Reference*
- *Drawing Generation Best Practices*
- *Parametric Modeling Best Practices*

Creating an Assembly Drawing Document

CAMU simplifies the design and management of projects and assemblies. It also enables you to create engineering drawings that describe whatever model instances are presently visible in the active Adrawing.

You have no write access to viewed but inactive models. You can, however perform associative hidden line removal. You can associatively dimension to viewed model instances, as well as to HLR. You can also shade the display of model instances currently viewed in the active Adrawing.

Please note: Associative HLR and associative dimensioning are optional. You can specify nonassociativity using property sheet options.

You can create assembly engineering drawing using associative HLR (Hidden Line Representation) and associative dimensions. Do this in the Explicit assembly environment. You can insert annotation geometry such as labels, balloon callouts, text, notes, and symbols into the active Adrawing to detail the viewed assembly. You can output the display to plotters and printers.

Please note: Do not detail the assembly in the `DEFAULT` Adrawing. Rather, use `DEFAULT` simply as a springboard into the assembly. Detail the assembly from in an appropriately named Adrawing.

Significance of the Active Adrawing

The Adrawing provides:

- A way to view selected model instances at a particular orientation and scale, properly located with respect to other models in the assembly
- A placeholder for newly inserted assembly items, such as dimensions, text, notes, callouts, and so on, resident in the active Adrawing
- A display tool for HLR and shading of viewed model instances

Generating Hidden Line Representation (HLR)

You can generate associative or nonassociative HLR in the CADDS Graphics window, using the HIDE OBJECT and UPDATE HLRIMAGE commands. This is the suggested method for engineering drawings.

You can optionally generate nonassociative HLR using the HLREDGE and HLRNOEDGE rendering options in the CHANGE RENDERVIEW commands available with CADDShade II functionality.

Please note: The HIDE OBJECT/HIDE HLRIMAGE commands and CADDShade II are only available in the Explicit assembly environment but valid for parametric and/or explicit viewed model geometry.

You can use one of the following methods to produce an assembly drawing document.

- HIDE OBJECT is the preferred technique for producing an assembly drawing document. Although it performs more slowly than CADDShade II, the graphic entities generated are associative (optionally) and in vector format, resulting in clear graphics for the assembly drawing document. HIDE OBJECT is performed in the Graphics window.
- Perform graphical hidden line removal using the HLREDGE and HLRNOEDGE rendering options in the CHANGE RENDERVIEW or RENDERVIEW commands. The hidden line removal performed in this method is faster but limited in quality and accuracy as compared to the analytical hidden line removal performed using the HIDE OBJECT command.

Creating an Adrawing Document

Before you perform hidden line removal, shading, or detailing of the assembly, do the following:

1. Activate a new Adrawing in the CAMU Explicit environment.
2. Define views at an appropriate scale, orientation, and location.
3. View on the model instances you want to use.

Example

1. Create a new Adrawing. Use a blank sheet (no views predefined) or a company standard format sheet (such as one containing your company logo, title block, and so on).

2. Define the desired views at the desired orientation, scale, and location on the Adrawing. For example, define a front view at half scale, locate it, and define the view clipping boundaries.
3. Use the View Component property sheet to view the desired component instance's model geometry.
4. Define any additional views. For example, if you have defined a front view, fold a right and top view off of it using viewed geometry from the front view.

Please note: To learn about manual view definition, including folding views, see the *Design and Drafting User Guide and Menu Reference* or the DEFINE VIEW online command help.

To generate HLR, see "Generating HLR Using HIDE OBJECT" on page 12-5.

To insert dimensions and annotation, see "Dimensioning and Annotating" on page 12-21.

Generating HLR Using HIDE OBJECT

This section describes how to perform hidden line removal (HLR) on assemblies made up of components associated with solid models. HLR consists of three command options that are accessible from the Hidden Line Removal command palette.

Using HIDE OBJECT, CADDS generates limbs and surrogate curves to accurately reflect the HLR appearance. You can create associative or nonassociative HLR. You can associatively dimension to the HLR curves. Use the UPDATE HLRIMAGE command to update the HLR display (if objects are resized, relocated and so on) and its associative dimensions. Or recreate the HLR, while maintaining dimensional associativity, using the HIDE OBJECT command again. Use the UNHIDE OBJECT command to restore hidden lines.

Remove Hidden Lines enables you to erase hidden lines and generate silhouettes on objects to produce a complete three-dimensional image. You can hide the lines in one view or all views. You can also blank the hidden lines or show them as dotted lines (see “Removing Hidden Lines” on page 12-6).

Restore Hidden Lines completely restores the hidden lines of the object. You can restore lines for all objects or only those that you select. You can also restore lines and keep the limbs created by Remove Hidden Lines (see “Restoring Hidden Lines” on page 12-15).

Update HLR Image enables you to update the present HLR image based on parametric modeling changes to the original geometry, relocation of the original geometry, and so on. Associative dimensions affected by the HLR update are regenerated and updated (see “Updating the HLR Image” on page 12-16).

HLR Commands

You can also use the following commands:

- Remove hidden lines using the command HIDE OBJECT
- Restore hidden lines using the command UNHIDE OBJECT
- Identify changes using the command MARK CHANGES
- Update the HLR graphics using the command UPDATE HLRIMAGE

For more information on hidden line generation and manipulation, see the *Hidden Line Removal (and AEC HLR) User Guide and Menu Reference*.

Removing Hidden Lines

Removing hidden lines reduces the visual complexity of an object and gives it a three-dimensional appearance. This enables the natural visualization of solid models by blanking those features that would be invisible or hidden from the viewer if the object were solid.

If the HIDE OBJECT command is applied to previously HLRed entities, it preserves the associative dimensions of the Limbs or Surrogates. Similarly, if the dimensions are applied to the viewed part entities, HIDE OBJECT preserves the dimensions for those curves which are still visible after the execution of the command.

Please note: When you perform an operation for hidden line removal, a warning message appears if the selected entity is not a properly formed solid. For viewed entities, this message displays the name of the part or component, its instance name (ID Name), and its Master Index Pointer (miptr) in the original part as follows:

```
Warning: Error in reading solid entity 324785.
Miptr 324785 is miptr 62641 in the model VALVE, component
valve_1.
```

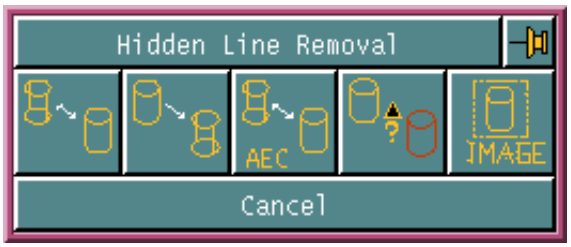
If dimensions are applied to the Surrogates in HLR Image, the UNHIDE OBJECT command preserves these dimensions and they are applied to the corresponding viewed part curves. Dimensions applied to Limbs are not transferred.

To Access HLR

1. Click the Render icon (paintbrush) on the Environment Global area of the top bar.

2. Choose Hidden Line Removal from the pulldown menu.

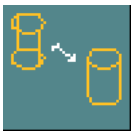
The Hidden Line Removal command palette appears.



The following steps describe how to use Hide Object menus to perform hidden line removal on a crankshaft assembly.

1. Click the Render icon (paintbrush) from the top bar.
2. Choose Hidden Line Removal from the pulldown menu.

The Hidden Line Removal command palette appears.



3. Select the Remove Hidden Lines icon from the palette.

The Remove Hidden Lines property sheet appears as follows:

Figure 12-1 Remove Hidden Lines Property Sheet

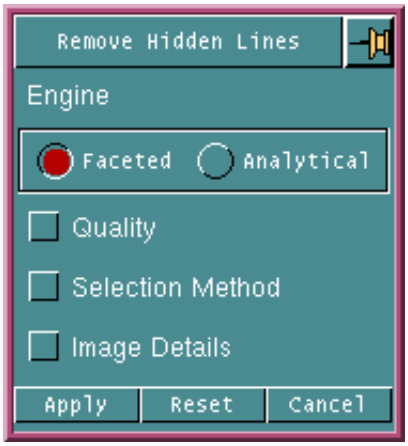
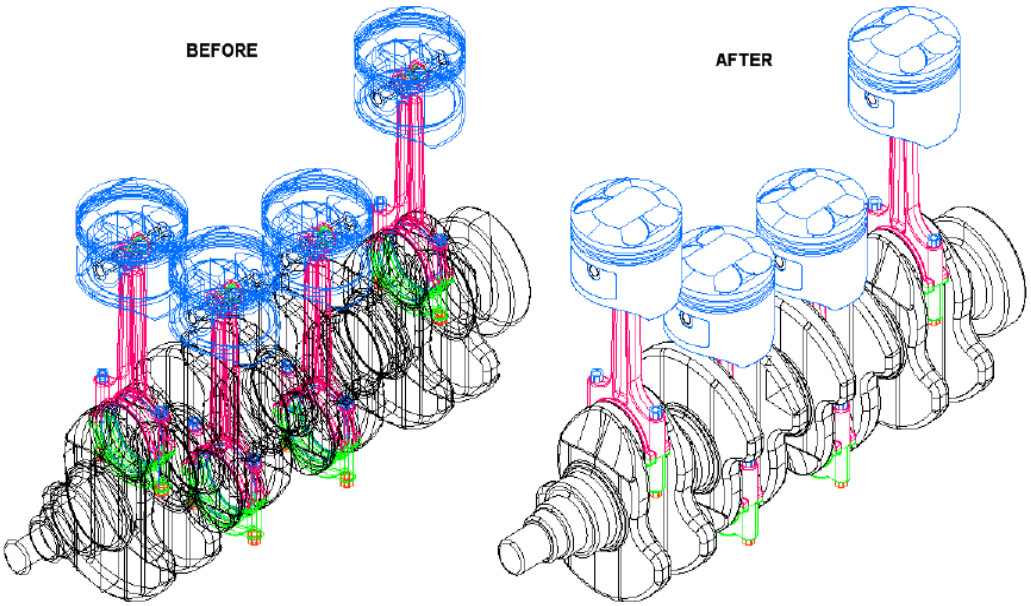


Figure 12-2 Crankshaft Assembly Before and After the Hide Object Command



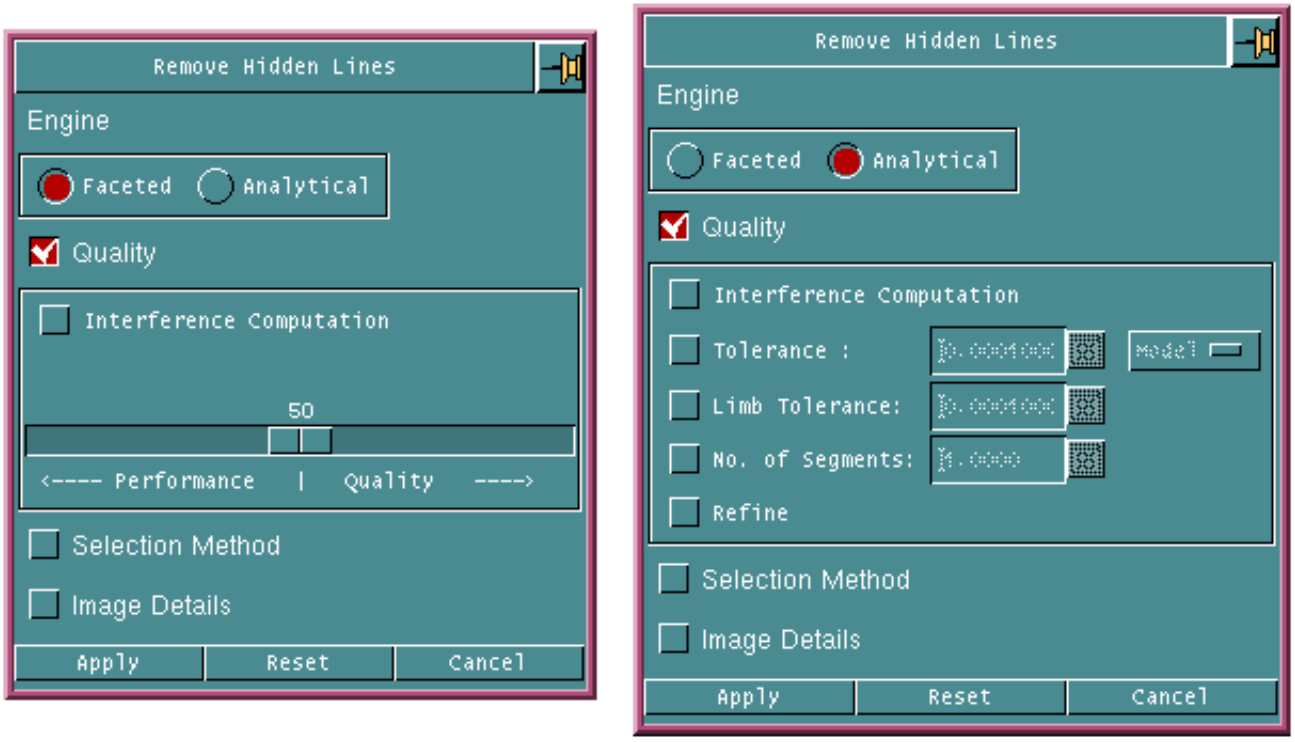
The following steps describe the options available to you for removing the hidden lines:

1. Select the engine that you want to use to remove the hidden lines.
   - Faceted to use the Faceted engine.
   - Analytical to use the Analytical engine.

Please note: While using the Faceted engine, you can interrupt the HLR operation by pressing ESC+Q while the HIDE OBJECT FACETED command is running. All options and modifiers common to the Faceted and Analytical engine are identical in function.

2. Select Quality to view the controls related to the quality of the results of HLR. The Faceted and Analytical modes of the Remove Hidden Lines menu appear as shown respectively:

Figure 12-3 Quality Option in Faceted and Analytical Mode



- Select Interference Computation to obtain better HLR results when the objects interfere with each other. Choose this option in the Analytical engine mode to divide all the curves into an independent number of segments. In the Faceted mode, facets are created instead of segments, and their number depends on the quality factor you select.

Please note: This option can decrease the processing speed.

Associative HLR is valid for all viewed model geometry (VIEW COMPONENT). Geometry need not exist physically.

- Select Intersection Curves to display the intersection curves while using the HIDE OBJECT command. This option and its submodifiers are available only if the Interference Computation option is selected first. Selecting this option automatically enables interference. This option involves a Boolean operation between the interfering objects and may take a significant amount of time for complex and large assemblies and parts. This option and its submodifiers are available only if the Interference Computation option is selected first. The following modifiers are available for this control:
- Layer: Specifies the layer in which the HLR intersection curves are displayed. If the layer is not specified, the active layer value is used.
- Tol: Specifies tolerance to be used for the intersection process. If the tolerance is not specified, the default tolerance value is used. This submodifier is not available in the Faceted engine mode.
- The slidebar available when the Faceted engine and Quality option is selected can be used to choose the quality of the HLR. The higher the quality, the more accurate the result and slower the calculation.

Please note: The modifiers Tolerance, Limb Tolerance, No. of Segments, and Refine are available only in the Analytical engine mode.

c. Tolerance : Use this option to determine whether two curves occupy the same place in a view. The default value is 0.0001 inches. In the field beside Tolerance you can specify the tolerance value either in Model units or in Draw units:

The tolerance value controls the following factors:

- Curves and Space: The smaller the tolerance, the more curves will be occupied in the given space.
- Limb accuracy: The smaller the tolerance, the more accurate the limbs are.

d. Limb Tolerance : Use this option to specify the tolerance factor in units for generating limbs. The default value is derived from the Tolerance option.

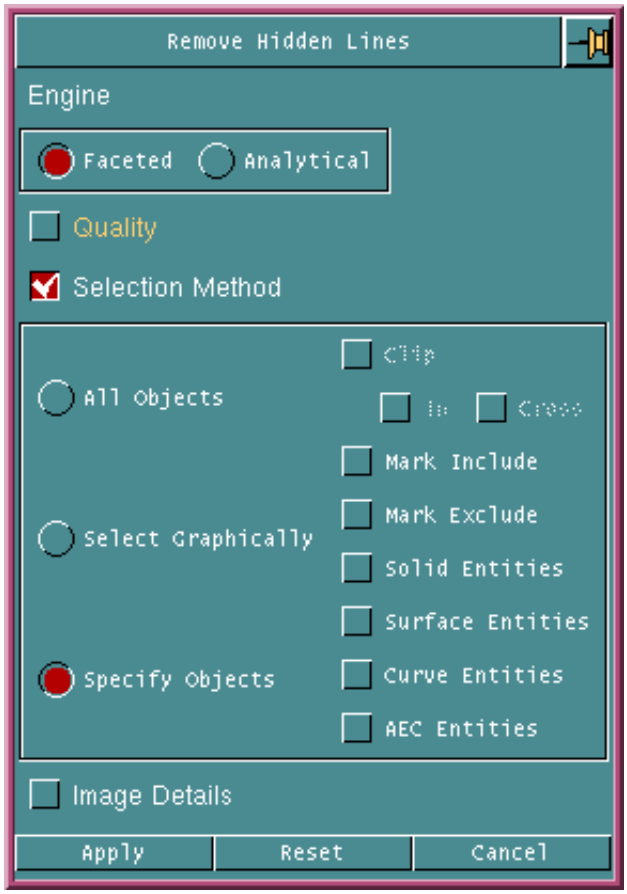
Warning

Using Limb Tolerance increases the size of the database and negatively affects the quality of the displayed object.

e. No. of Segments: Use this to select the number of segments to be generated. The default value is 1.000.

f. Refine: Use this option when the HIDE OBJECT command does not produce accurate Hidden Line Removal. This option provides better quality results of an HLR image. In the Faceted engine mode, the Quality factor may be adjusted to achieve similar results.

Figure 12-4 Selection Method Options



3. Enable Selection Method to view the available options. These options are the same for Faceted and Analytical modes. The options are as follows:

a. Select All Objects to remove hidden lines from all the objects that are shown in the views you choose. This is the default option if you select the Selection Method option. In the Faceted mode, All Objects is the default option, even if you do not select the Selection Method option. In the Analytical mode, All Objects is not the default option.

b. Use the Clip option to remove hidden lines only from those entities that lie within or cross the selected view boundaries. This option has the following modifiers:

- In: Removes hidden lines from entities inside the view clip volume.
- Cross: Removes hidden lines from entities that cross the view clip volume.

- In and Cross: Select both the options to remove hidden lines from the entities that lie within and cross the selected view boundaries. Alternatively, perform the same operation by clearing the In and Cross options.

c. Choose the Mark Include option to remove hidden lines from marked entities only.

d. Choose the Mark Exclude option to remove hidden lines from unmarked entities only.

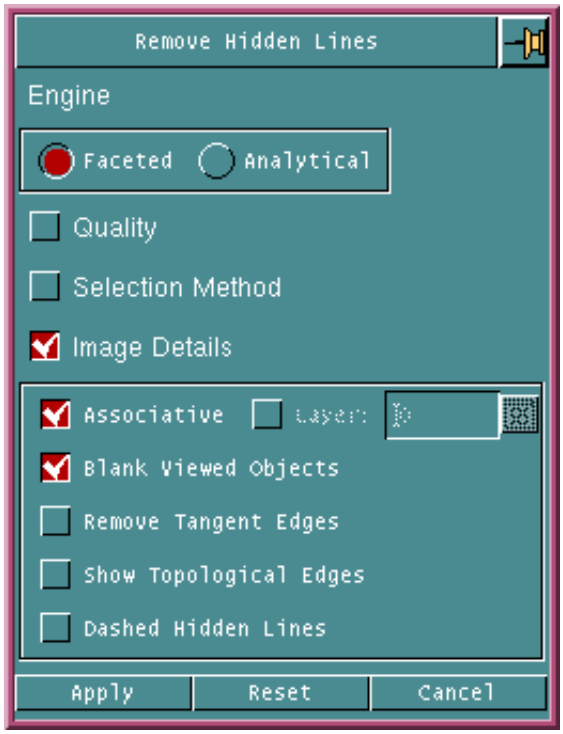
e. Choose Select Graphically to remove hidden lines from the objects that you select graphically

f. Select Specify Objects to remove hidden lines from all the entities of a specified type. You can choose one of the following entities:

- Solid Entities: Removes hidden lines from all visible solid entities.
- Surface Entities : Removes hidden lines from all visible surface entities.
- Curve Entities: Removes hidden lines from all curved entities.
- AEC Entities: Removes hidden lines from all visible AEC entities.

4. Select Image Details to view the related controls. These controls are common for both Faceted and Analytical modes:

Figure 12-5 Image Details Options



a. Associative: The HLR image that is created is associative; that is, whenever you move a viewed component, all the HLR curves and limbs corresponding to the component also move. This is the default. When you do not select this option, CADDS applies the NONASSOCIATIVE modifier to the command HIDE OBJECT and the image created is nonassociative. For a nonassociative image, the HLR curves and limbs do not move when the underlying component is moved. You can create a nonassociative image in the CAMU environment only.

b. Layer: This is a suboption to the modifier NONASSOCIATIVE. With this option, all the nonassociative curves can be put on the selected layer.

c. Blank Viewed Object: Use this option to override the default which causes view parts or assemblies to be blanked during the Remove Hidden Lines process. If you select this option, you must manually blank the viewed parts in order to obtain the correct image.

d. Remove Tangent Edges: Choose this option to suppress all the tangent edges for the selected solids. On selecting this option, CADDS identifies all the tangent edges in all the selected objects and does not display them in the final result.

e. Show Topological Edges: When you use the HIDE OBJECT command CADDS blanks the Topological Edges by default. Choose this option to display the omitted Topological Edges.

f. Select the Dashed Hidden Lines to display the hidden lines as dashed lines and select Dash Length and specify it in absolute drawing units. The default value is 0.0001. Select Space Length and specify the space between dashes in absolute drawing units. The default value is 0.0001.

5. Click Apply.

Please note: HIDE OBJECT inserts new wireframe entities into the active Adrawing. These new entities display the HLR representations of the viewed solid models, which are blanked. The blanked state of the underlying model graphics is temporary. Use the Unview option to permanently turn off the display of the blanked models.

6. At the view prompt, select the view or views on which you want to perform the HLR operation.

7. Select Objects to Hide to obtain the CADDS model entity prompt.

8. Select the viewed models for HLR generation.
9. Select Done.

CADDS generates HLR for the selected models in the selected view or views.

You can associatively dimension to HLR. See “Inserting Dimensions and Annotation” on page 12-22 for details.

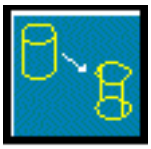
Assembly Structure Display Results

When you generate HLR using HIDE OBJECT BLANK, the affected components are prefixed with the appropriate icons in the assembly structure. For details see the section, Assembly Tree Window.

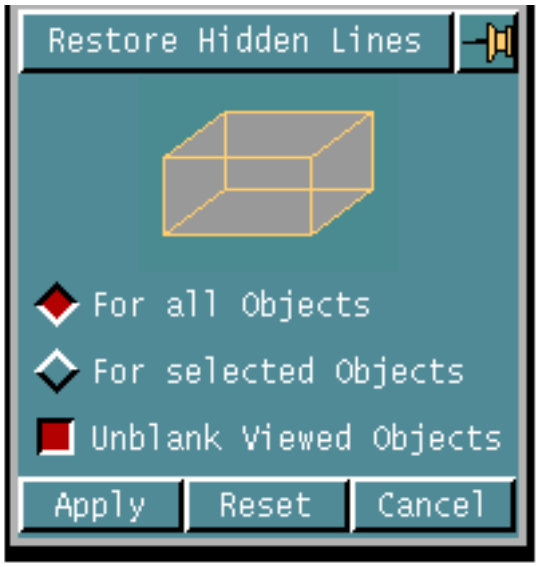
Restoring Hidden Lines

The following steps describe how the Restore Hidden Lines icon restores (unblanks) the hidden lines of all or the selected objects.

1. Select the Render icon (paint brush) from the top bar.
2. Select Hidden Line Removal from the pulldown menu.
3. Select the Restore Hidden Lines icon from the Hidden Line Removal command palette.



The property sheet appears as shown below:



4. Choose one of the following options.
   a. Choose For all Objects to restore all objects in the specified view, including the objects that are completely hidden. Graphically select the views for which you want to restore the hidden lines.
   b. Choose For Selected Objects to restore hidden lines for selected objects only. Graphically select the views for which you want to restore the hidden lines.
   c. Choose Unblank Viewed Objects to unblank the viewed models that were blanked by an earlier HLR process.
5. Select Apply.

Please note: The Restore Hidden Lines property sheet is an immediate action property sheet. After you have selected Apply, you cannot abort the command unless you select Abort from the Utilities command palette (right mouse button in Graphics window) or enter <Control e>.

Updating the HLR Image

To enable associative HLR, set the following variable in your `.caddsrc-local` file:

```
setenv CADDSASSOC_UID "yes"
```

Associative HLR does not update automatically when the original geometry is modified. The UPDATE HLRIMAGE command updates previously generated, associative HLR curves. It has no effect on nonassociative HLR.

UPDATE HLRIMAGE rebuilds the HLR. It recomputes curves (limbs and surrogates) to reflect any changes that have occurred since the previous HLR image was generated. It also updates any associative dimensions to or between the affected HLR.

Associative dimensions to or between associative HLR are also updated by the HIDE OBJECT command.

This functionality can be used:

1. In an active model where HLR is only based on geometry that is physically resident in the active model. This is valid when parametric changes are made to the entity or when limbs are regenerated.
2. In CAMU where HLR is based on viewed model geometry that is not physically resident in the active Adrawing. This is valid when one or more viewed model instances (VIEW COMPONENT) undergo a change (size, shape, location).

Generating HLR Using CADDShade II

The CADDShade application functionality is now displayed in the main graphics window. The optional Picture window is no longer available. You can view Parametric models in the Explicit environment in part mode or in CAMU. You can access CADDShade settings from the Setup menu on the top bar and invoke commands from the Interactive View Control option on the VIEW menu. You can also generate nonassociative HLR of the model using the appropriate rendering options in the RENDER VIEW and CHANGE RENDERVIEW commands.

CADDShade functionality generates hidden line representation very quickly, including, if required, drawing annotation such as dimensions, notes, drawing border, and so on. However, the resultant raster graphics are limited in quality and at handling component intersections when compared with those generated using HIDE OJECT (see “Generating HLR Using HIDE OBJECT” on page 12-5).

For more information refer to the *CADDSHADE II User Guide*.

File Structure and the Shade/Hide Picture File

To use the CADDShade II functionality, each viewed model in the assembly must have the `_gr` file and the tessellation data stored in the `_gr` file must be in sync with the model.

Preparing to Use CADDShade II

For the CADDShade program to recognize and perform the shaded or hidden appearance of a viewed model in the assembly, the viewed model must have a current `_gr` file containing the drawing and model graphics data along with the tessellation data. If the _gr file of your model is missing, the VIEW COMPONENT command dynamically generates the graphics data for that model by reading the `_pd` file of that model . However, this can adversely impact the performance of the VIEW COMPONENT command.

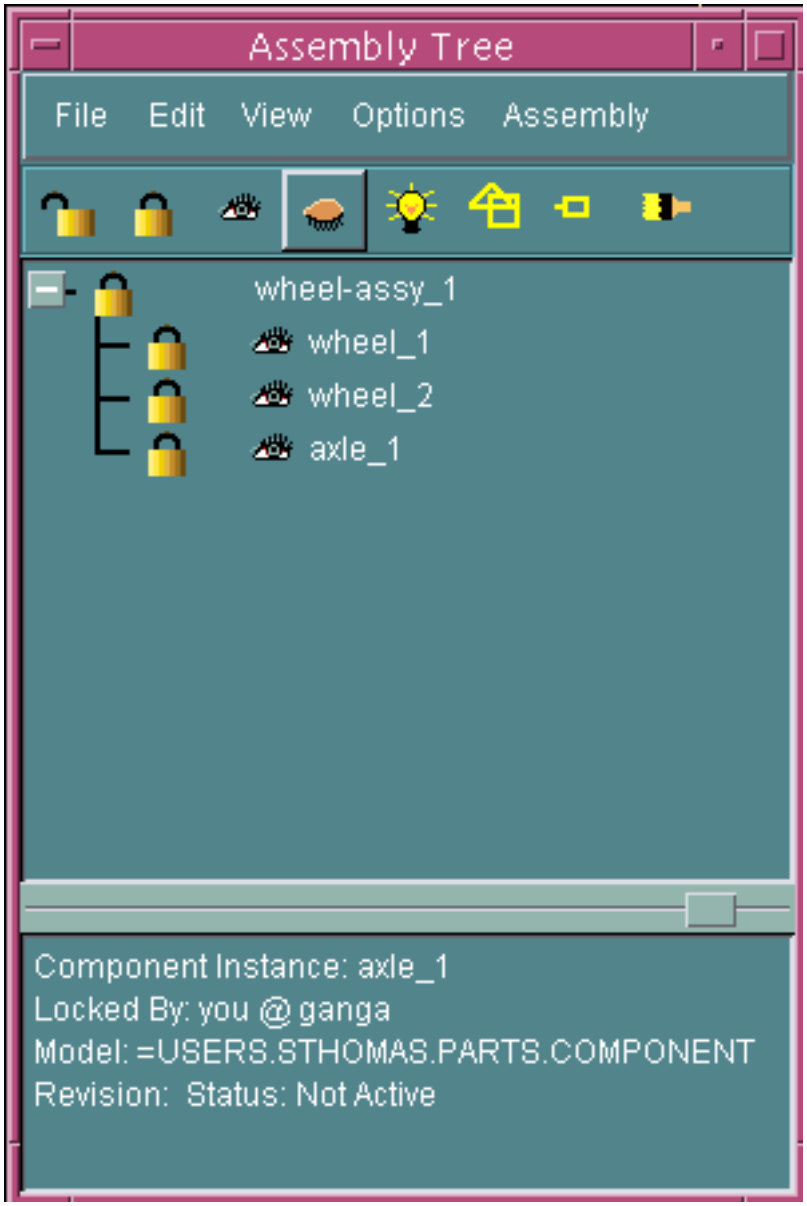
For more information refer to the *CADDSHADE II User Guide*.

Generating CADDShade II HLR

This section shows you how to generate CADDShade HLR of viewed components.

You can shade or hide viewed components using CADDShade. Your graphics display output appears in the main graphics window. Follow these steps:

1. Activate an Adrawing.
2. View all desired components.

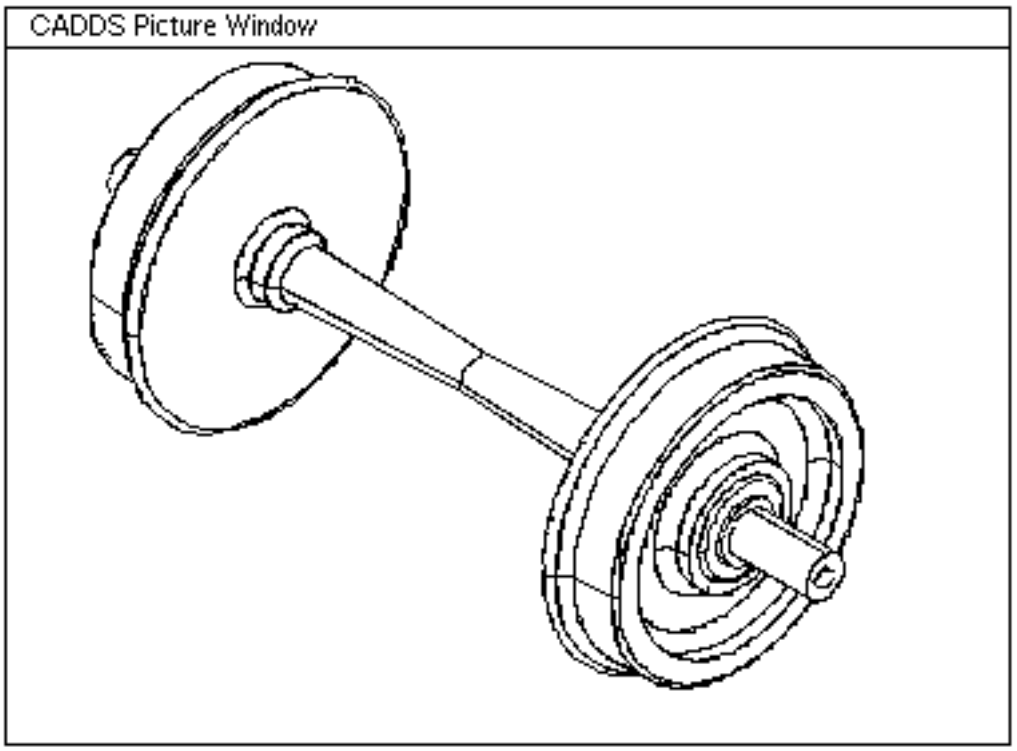


3. Establish hidden line defaults for the assembly.

4. Perform CADDShade Hidden Line generation using any of the following commands.

```
CHANGE RENDERVIEW HLREDGE
CHANGE RENDERVIEW HLRNOEDGE
RENDER VIEW HLREDGE
RENDER VIEW HLRNOEDGE
```

CADDSHADE HIDDEN LINE



For more information refer to the *CADDSHADE II User Guide*.

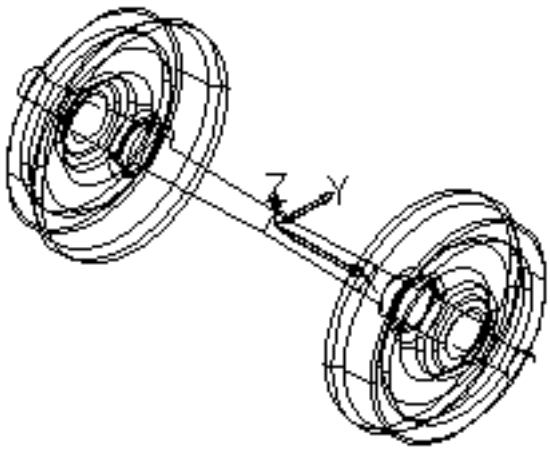
Dimensioning and Annotating

You can create associative annotation in an Adrawing of the following types:

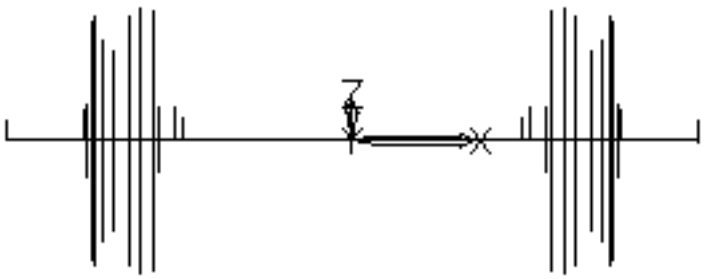
- Hidden line representation (HLR)
- Dimensions
- Hatch
- Sections

This section describes how to create dimensions to viewed model geometry and HLR in the active Adrawing. It does not discuss the variety of dimensioning styles and detailing options available within CADDS.

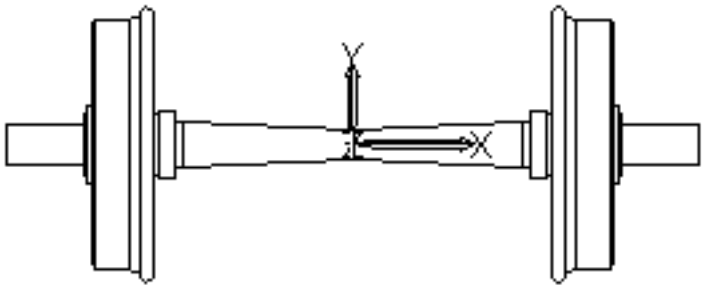
For more information about CADDS detailing options, see the *Design and Drafting User Guide and Menu Reference*.



View orientation has been changed to orthographic front for the wheel assembly. The `wheel_1`, wheel_2, and `axle_1` component instances are viewed on.



Associative HLR has been generated.



Inserting Dimensions and Annotation

You can insert associative dimensions and labels, text, and other annotation to describe viewed model geometry in the active Adrawing. When the viewed model geometry to which you have dimensioned is modified or moved, the associative dimensions update accordingly.

You can associatively dimension to HLR geometry. This specifically pertains to limbs and surrogates generated by the HIDE OBJECT command. To accommodate this enhancement, use the UPDATE HLRIMAGE command. This regenerates the HLR and updates dimensions accordingly.

Be sure to activate an Adrawing to contain the dimension and annotation entities. These items are visible only when this particular Adrawing is active.

Please note: Entities that are physically resident in the Adrawing, such as annotation and HLR, are only visible when that Adrawing is active.

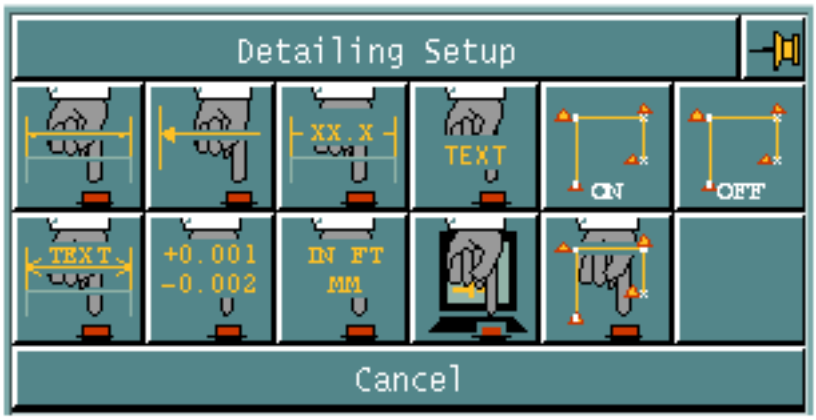
Establishing Associativity Default

Establish dimensions as associative before their insertion.

Perform the following steps to establish associative dimensioning defaults in the active Adrawing.

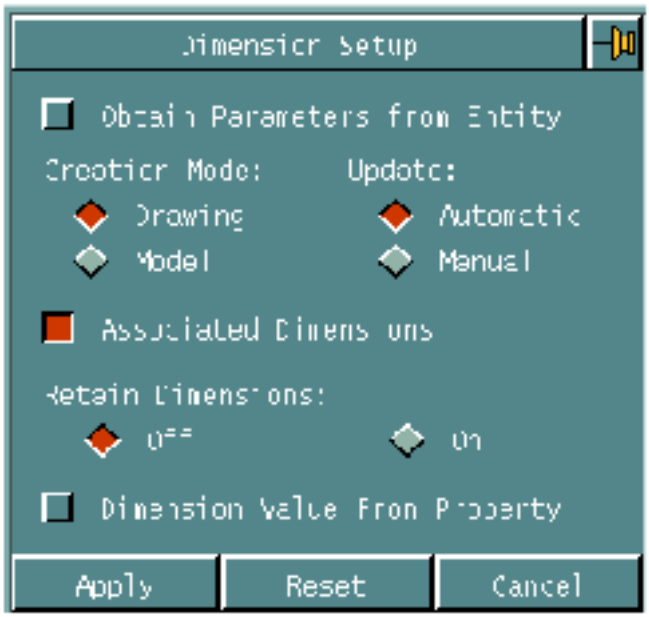
1. Select Setup from the top bar.
2. Select Detailing from the pulldown menu.

The Detailing Setup command palette appears.



3. Select the Dimension Setup (top left) icon.

   The Dimension Setup property sheet appears.

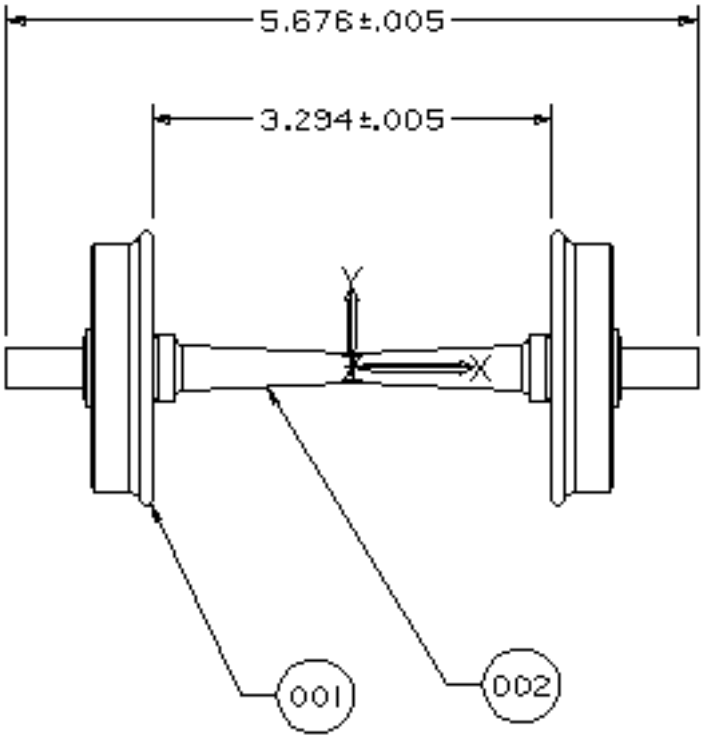


4. Ensure that the Associated Dimensions option is enabled (appears red).
5. Click Apply.

Inserting Dimensions

1. Change your task set to Drafting. To do this, select the task set name (COMP/ASSY or whatever task set is active). Select Drafting from the resultant pulldown menu.
2. Insert dimensions as needed.
3. Insert annotation as needed.

Please note: See the *Design and Drafting User Guide and Menu Reference* for instructions on how to insert dimensions and annotation.



Please note: If a particular surface has more than one limb, all its dimensions become nonassociative and are marked up.

If a part corresponding to a component is renamed or deleted, dimensions associated with the viewed component are lost.

You can optionally change associative dimensions to nonassociative using the CHANGE DIMENSION NONASSOCIATIVE command. However, you cannot associate a nonassociative dimension with a viewed entity using the CHANGE DIMENSION ASSOCIATIVE command.

Regenerating Dimensions

- If you blank a component using the BLANK COMPONENT command, the dimensions associated with the blanked component remain as they were before blanking.
- If you unview a component using the UNVIEW COMPONENT command, the dimensions associated with the unviewed component appear in red along with an encircled slash symbol. This state of the dimension indicates its partial disassociation with the unviewed component. After using the UNVIEW COMPONENT command, if you view the same unviewed component using the VIEW COMPONENT command, the dimension retains its original color. The red color of the dimension changes to the original color and the encircled slash symbol of the dimension is removed.
- If you change a component using the CHANGE COMPONENT command and if the topology of the new component differs from the topology of the original component, the dimensions associated with the changed component are disassociated and are marked. If there is no difference between the topology of the new component and the original component, the dimensions associated with the changed component remain the same.
- If you remove a component using the REMOVE COMPONENT command, the dimensions associated with the removed component are disassociated.

Detailing Models in CAMU

When you activate a model, CAMU automatically activates a drawing in the model database named `MDRAW`. Do not detail models in their `MDRAW` drawing. You should never use a model's `MDRAW` drawing as an engineering drawing. Views that you manually define on the `MDRAW` are never saved. The purpose of the `MDRAW` is only to facilitate visually seamless movement as you move between active models. It acts like a mimic of what was visible in the last active part.

Detail a model in any drawing other then `MDRAW`. To activate another drawing in the active model, perform the following menu sequence:

Adrawing>Drawing>Activate New or Activate Old

Once active in the drawing, define views as you normally do when creating Explicit detail drawings. For example, define a Front view, and fold a Top and Right view from it. Insert associative dimensions, and so on.

Please note: All models that were visible (VIEW COMPONET) in the previously active part (model or Adrawing) are presently visible. To display only

the model geometry that is physically resident in the active part, unview (VIEW COMPONENT) the entire assembly structure.

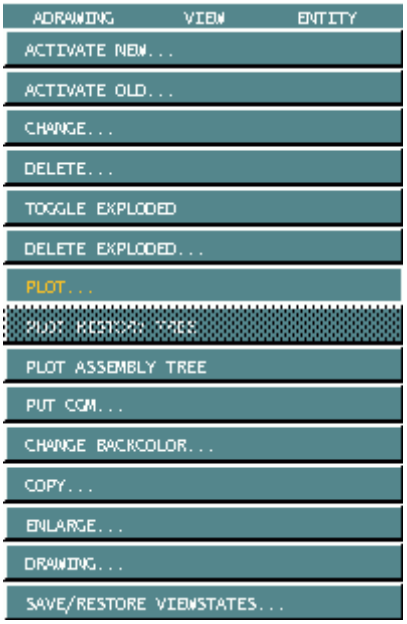
Plotting

You can plot the contents of the Graphics window. You can also plot the contents of the Assembly Structure window.

Plotting the CADDS Graphics Window

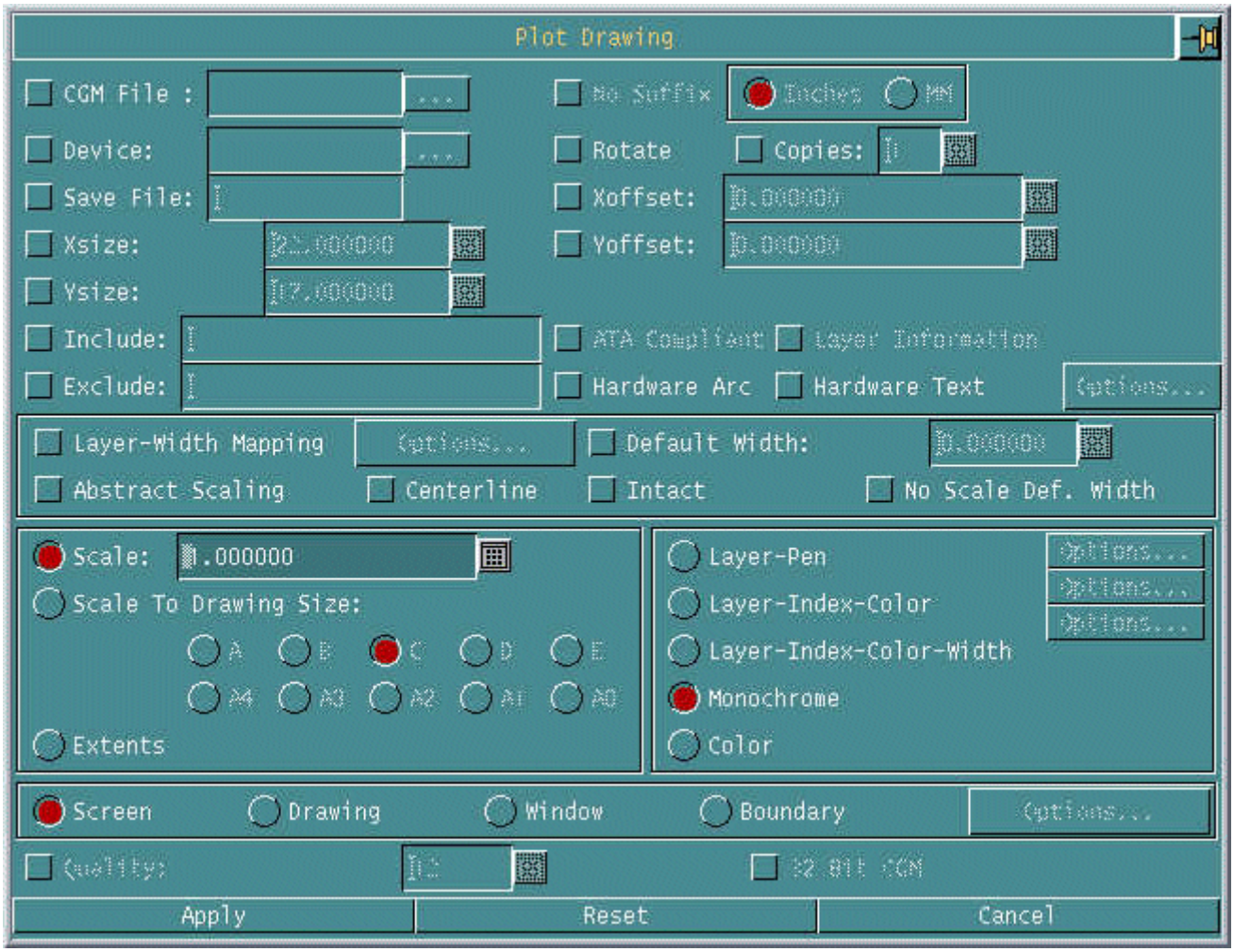
You can plot the contents of the CADDS Graphics window to a dot matrix, laser, or electrostatic printer or plotter.

1. Select Adrawing from the top bar.
2. Select Plot from the resultant pulldown menu.



The Plot Drawing property sheet appears.

3. Specify desired options and click Apply.



Please note: For more information, see PLOT DRAWING in the online command help and the *PTC Plotter Drivers Administration Guide*.

Plotting the CADDShade Image

To send the display of shaded or hidden line removed graphics to a printer or plotter, issue the following command in the CADDS Text window:

```
PLOT DOT HIDE
PLOT DOT SHADE
```

Please note: You can only plot images that have been rendered using the CHANGE RENDERVIEW command and not the RENDER VIEW command.

For more information, see PLOT DOT in online command help.

Plotting the Assembly Structure

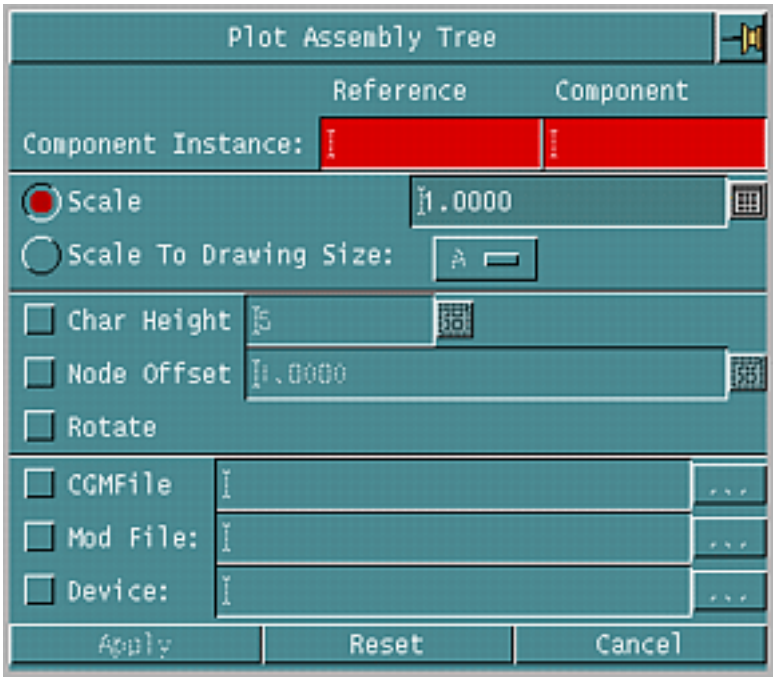
You can plot the assembly structure of an active assembly using the Plot Assembly Tree option. You can plot the assembly structure directly to the plotter device or save the vector output to a standard CGM file. You can choose to plot a component node (sub-assembly) of the assembly structure along with its dependent nodes.

Set the value of the `CVUPS_TEXT_OUT` environment variable to "`yes`" in the `.caddsrc` or `.caddsrc_local` file, to enable plotting of hardtext.

The Plot Assembly Tree option is available in both the Parametric and Explicit Assembly environment.

To plot the assembly structure,

1. Choose the Adrawing option on the top bar.
2. Choose the Plot Assembly Tree option on the Adrawing menu. The Plot Assembly Tree property sheet appears.



You can specify the options on the Plot History Tree property sheet as:

- Reference: Specifies the name of the reference assembly (assembly instance). Once specified, you can plot the assembly structure of a Component within the reference assembly. Leave this field blank if you want to plot a component in the active main assembly.
- Component: Specifies the component instance to plot within either the active main assembly or the reference assembly that you specify.
- Scale: Specifies the scaling factor for the output. The default scaling factor is 1.
- Scale To Drawing Size: Specifies to scale the output to a standard drawing size. Choose the required standard drawing size. The default size is A4.
- Char Height: Specifies the character height of the text of the assembly structure node.
- Node Offset: Specifies the offset value to use during plotting.
- Rotate: Specifies to rotate the assembly structure by 90 degrees before plotting.

- CGMFile: Specifies the file name of the output CGM file. Do not specify the CGM extension for the output file.
- Mod File: Specifies the file that contains the print and plot attributes. Use this file to print the additional attributes of the assembly structure nodes. The additional attributes for the assembly structure node are:
  - full part name of model associated with the component
  - active status of the component instance
  - db_id of the component instance
  - name of the component

  You can also specify the plot attributes using the mod file, instead of specifying them on the Plot Assembly Tree property sheet.

Please note: The plot attributes that you specify in the mod file override the options you select on the Plot Assembly Tree property sheet.

A copy of the mod file is available at:

`cadds/scripts/templates/xmlcamutree.cfg`

Modify this file as per your requirement and use it to generate the plot output.

- Device: Specifies the output plotter device. The plotting device name appears in the Device field. To use another plotter, select the required plotter from the available list.

Please note: If you do not specify the CGMFile and the Device options, a file named `tree.cgm` is created automatically in your CADDS create directory. If you specify the Device option without the CGMFile option, the `tree.cgm` file is created in the directory specified by the environment variable CV_AW_TMP_DIR. In both cases any existing `tree.cgm` file is overwritten.

3. Click Apply on the Plot History Tree property sheet to generate the plot output.

You can also use the supplied CVMAC `tree` routine to generate entities in the CADDS Graphics window that reflect the contents of the assembly structure (see "Generate Assembly Structure Entities" on page 11-19).

Once generated, plot using the technique described in "Plotting the CADDS Graphics Window" on page 12-27.

Plotting the Part History Tree

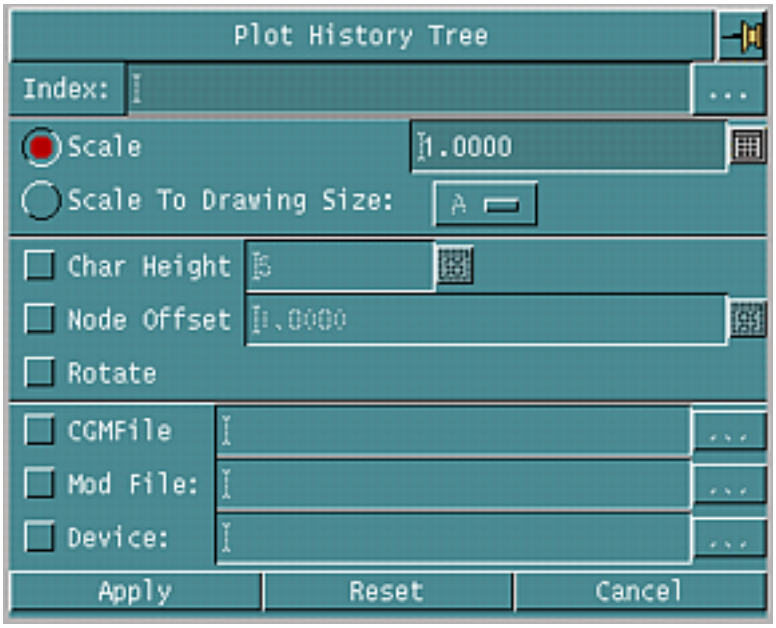
You can plot the history tree of an active part using the Plot History Tree option. You can plot the history tree directly to the plotter device or save the vector output to a standard CGM file. You can choose to plot a node of the history tree along with its dependent nodes.

Set the value of the `CVUPS_TEXT_OUT` environment variable to `yes` in `.caddsrc` or `.caddsrc_local` file, to enable plotting of hardtext.

Please note: The Plot History Tree option is available only in the Parametric Assembly environment.

To plot the history tree,

1. Choose the Drawing option on the top bar.
2. Choose the Plot History Tree option on the Drawing menu. The Plot History Tree property sheet appears.



Specify the options on the Plot History Tree property sheet as:

- Index: Specifies to plot the history tree for the selected history tree node along with its dependent nodes. If you do not specify any Index value, the complete history tree is plotted.
- Scale: Specifies the scaling factor for the output. The default scaling factor is 1.

- Scale To Drawing Size: Specifies to scale the output to a standard drawing size. You can choose the required standard drawing size. The default size is A4.
- Char Height: Specifies the character height of the text of the history tree node.
- Node Offset: Specifies the offset value to use during plotting.
- Rotate: Specifies to rotate the drawing by 90 degrees before plotting.
- CGMFile: Specifies the file name of the output CGM file. Do not specify the CGM extension for the output file.
- Mod File: Specifies the file that contains the print and plot attributes. Use this file to print the additional attributes of the history tree nodes. The additional attributes for the node are:
  - command tolerance
  - command version
  - command Cplane
  - suppress state of the command
  - save state of the command

  You can also specify the plot attributes using the mod file, instead of specifying them on the Plot History Tree property sheet.

Please note: The plot attributes that you specify in the mod file override the options you select on the Plot History Tree property sheet.

A copy of the mod file is available at:

cadds/scripts/templates/xmlhisttree.cfg

Modify this file as per your requirement and use it to generate the plot output.

- Device: Specifies the output plotter device. The plotting device name appears in the Device field. To use another plotter, select the required plotter from the available list.

3. Click Apply on the Plot History Tree property sheet to generate the plot output.

Glossary

Adrawing

A special CADDS part, used by a CAMU assembly database, to optionally view some or all of the models that comprise an assembly. It is essentially both a CADDS part and drawing. The Adrawing can contain representation of viewed model geometry (VIEW COMPONENT), views at any scale, orientation, and location, specific layer offset information, detail geometry such as dimensions, balloon callouts, crosshatch, parts list text, etc. It can also house HLR representation.

Each assembly has a minimum of one Adrawing (the `DEFAULT` Adrawing) and can have multiple Adrawings.

Entities that physically reside in an Adrawing are only visible when that Adrawing is active.

Each Adrawing can optionally contain two view states - standard (assembled) or exploded (alternate locations for viewed model geometry).

When you activate an Adrawing in CAMU, the part is activated, as well as a drawing named `DDRAW`.

Assembly database

The actual assembly file on disk. For example, when you activate an assembly named `061393`, the actual assembly database is an `_db` file resident in the directory `061393`.

Assembly instance

The root component of the reference assembly. See Reference instance.

Assembly structure

The hierarchical representation of the assembly or product as seen in the Assembly Structure window. It graphically represents the relationships between the piece parts and subassemblies that comprise the overall assembly.

Base component

The component with respect to which all other components are parametrically positioned. A base component is identified in order to assemble the selected parent component. There can be only one base component for each subassembly for which parametric positioning has been defined.

CPL

Acronym for Construction Plane. Also referred to as Cplane. The active CPL determines the current modeling coordinate system - X0Y0Z0 and orientation of the X, Y, and Z axes.

Child component

A component that is one level down from a parent component.

Component

A generic term for any one piece of the structure, shown as a small box. A component is also referred to as a node. It represents either a piece part, subassembly, or an assembly.

Component class

A unique name assigned to a component. Multiple instances of a component class can exist in the assembly structure. Operations performed on a member of a component class are reflected in all other instances of that class, both in the structure and the model representation of the class.

Component instance

One single occurrence of a component class.

Drawing

A workspace that supports any number of views and that contain various annotation items such as dimensions, hatch, text, etc. Drawings reside within a part database. A CADDS part can contain many drawing files. See Part.

Explicit Graphics File (_gr)

A file that stores the drawing and model graphics data along with the tessellation data. The `_gr` file allows you to see a part's graphics in the active drawing.

Get

The act of moving a copy of a file from a central database management repository, such as Optegra Vault, to your local disk.

HLR

Acronym for hidden line representation. HLR is created and manipulated using the HIDE OBJECT, UNHIDE OBJECT, and UPDATE HLRIMAGE Explicit commands.

LDM

See Local Data Manager.

Leaf node

A component that has no children. Leaf components are those typically associated to CADDS piece parts.

Local data manager

Start up screen that appears when you invoke CADDS. The LDM lists either CADDS parts, assemblies, or features databases. You can select and activate an existing database or create a new one.

Local disk

The disk attached to your machine. Parts on this disk cannot be viewed or modified by other users. If another user wants modify access to any parts signed out to another user he must go through the normal procedure of having the part put back to the Vault, then get the part to his local disk.

MSO

Acronym for Model Space Origin. MSO is the X0Y0Z0 of the seven supplied CPLs. MSO location and orientation is identical to the supplied Top CPL.

Main assembly

The active assembly whose root node is the leftmost in the assembly structure. This is the assembly activated using the Activate Assembly command.

Model

A CADDS piece part associated to a leaf node in an assembly structure. See Part. You can use a model to represent a weldment. You can also associate a model to a subassembly node in order to house outline geometry.

When you activate a model from within CAMU, the part is activated, as well as a drawing named `MDRAW`.

The terms model and part are often synonymous.

Model space origin

The datum X0Y0Z0 of a model. This is specifically the location at which the seven supplied CPL's coexist. Orientation of MSO is coincident with the Top CPL.

Nested reference assembly

A reference assembly that exists within a parent reference assembly.

Node

See Component.

Node switching

The process of changing from an active adrawing or model part to another Adrawing or model part in an assembly design session. This facility helps in modeling individual parts while referencing the geometry of other components of the assembly.

Parent component

A component that represents a subassembly (also called a subassembly node). A parent component must have a child component, and it can be a child of a higher-level component.

Part

A CADDS database that contains graphic and nongraphic information, any number of drawings, and any number of views on those drawings. See also Model.

Product structure

See Assembly structure.

Put

The act of moving a file from your local disk to a central database management repository such as Optegra Vault.

Read only disk

This houses work in progress assemblies (and models) available for concurrent viewing of project data. Any part which is checked out to your local disk and filed from the menus is 'copied' to this disk to allow other users access to view this part or Assembly.

Reference assembly

The Assembly that is created through the Add Assembly reference command. See Assembly instance.

Reference instance

The root component of a reference assembly. Also called an Assembly instance. Accessed using the command modifier AINAME.

Rigid assembly

An assembly in which the relative position of components, with respect to one another, does not change. A rigid subassembly behaves like a group.

Root node

The highest level component in the assembly structure. It is the leftmost component in the Assembly Structure window, representing the entire assembly. It is the highest component in the structure hierarchy. Each assembly can have only one root node.

If you attempt to activate (ACTIVATE MODEL) the root node (for the sole purpose of performing parametric interpart constraint operations) CAMU activates a model named root.

Subassembly node

See Parent component.

Tree

See Assembly structure.

View

A window into 3D model space at any scale, orientation, and location on a drawing. Views can be layer specific. A CADDS drawing can contain many views.

Viewstate

A viewstate contains the information about the layer offset, layermask and the view on or off state. each Adrawing contains it's own, unique viewstate. For example, one Adrawing may display (VIEW COMPONENT) all associated models in an exploded or alternate location state. Another may display (VIEW COMPONENT) one subassembly only in the standard or actual location state.

Note: CAMU is designed such that viewstate is never saved within a model `MDRAW` (ACTIVATE MODEL) but is always saved within an Adrawing `DDRAW` (ACTIVATE ADRAWING. This enables visually seamless Node switching.

Index

Symbols

.caddsrc file information 1-37, 2-3

A

Aborting a command 2-23
Accessing ATB 1-40
Accessing CAMU
  automatic 2-3
  from suntools 1-39
Accessing command 1-24, 1-31
  from Assembly Structure window 1-31
  from icon 1-24
Activate Model 8-3
  on global menu 1-24
Activate New Assembly property sheet 3-3
Activating a reference assembly 5-34
Activating an assembly 3-2
  existing 5-2
  new assembly 3-2
  view in graphics window 5-5
AD_LIST_CONSTRAINTS 2-5
Add component property sheet 3-9
Add Equation property sheet 9-14
Adding a component 3-9
  multiple instances 3-17
    component name 3-19
    example 3-18
    of a component class 3-15
    of a parent component 3-17
    of the same component 3-18
  to the assembly structure 3-7
Adding an assembly 3-21
  difference between adding and referencing 3-21
  existing 3-21
  prefix with 3-30
  reference 3-23
  referencing 3-24
  to the assembly structure 3-7
Adrawing 1-9, 5-5
  activating a model from 5-8
  Alternate and Exploded 5-5
  changing 10-11
  default Adrawing 3-2
  deleting 10-12
  menu selection 10-9
  moving from active to new 5-8
  significance 10-2
  standard view 10-7
  template 5-4
  view state 10-5
  viewing model in 10-5
Align 6-15
All Instances button 3-19
Alternate view 10-6
Ambiguous selection 6-8
Annotating 12-21
  an assembly 12-22
  associative update 4-2
  model geometry 12-21
anon variable setting 2-6
Apply on a property sheet 1-31
Archiving the assembly 2-22
Assemble component 6-3
  automatic 6-32
  manual 6-32

Assembly 1-7
  Explicit tasks set 1-24
  Parametric task set 1-24
Assembly drawing generation 6-2, 12-2
Assembly instance 1-8
  activating associated assembly 3-21, 5-4, 5-31
  creating 3-21
  editing external assembly 7-27
  filing edits 4-3
  modifying associated assembly 7-25
Assembly refreshing 5-36
Assembly Structure 1-2, 1-7
  blanking components in 7-29
  building 3-5
  changing 7-2
  changing appearance 7-3
  changing appearance of 7-29
  compressing components in 7-30
  creating guidelines 3-6
  example 1-10
  line between components 3-20
  naming convention 3-7
  obtaining information 7-3
  plotting 12-29
  relationship between components in 3-20
  structuring approach 3-5
  terminology 1-7
Assembly Tree 1-11
  shortcut menu 1-15
  top bar menu 1-19
Assist line 1-33
Associate Attribute Group property sheet 11-5
Associative Mass Point 5-22
  Merge Mass Point 5-24
Associative Topology Bus 1-40
ATB 1-40
ATB operations 1-40
Attribute 11-2
  assigning new name 11-16
  assigning value 11-11
  Attribute Editor window 11-11
  changing property 11-16
  creating 11-3, 11-8
  creating report with 11-19
  editing 11-11
  floating-point number in 11-10
  removing 11-17
  type 11-9
Attribute group 11-4
  associating with component 11-5
  creating 11-4
  listing 11-7
  removing 11-18
  removing association with component 11-6
Automatic mode, solving equation in assembly 9-12

B

Base component 6-17, 6-33
  behavior 6-33
  designation 6-34, 6-35
Blanking a subassembly 7-29
BOM CVMAC program 11-19

C

CADDS_MAX_PART 2-4
CADDSASSOC_UID 2-4
CADDSENABVP 2-3
CADDShade II 12-17
CALLBACK_SERVER, crash recovery 2-24
CAMU 1-2
  accessing function 1-28
  Adrawing options 1-22
  assembly options 1-22
  attributes task set 1-26
  component options 1-22
  crash recovery 2-24
  data management 2-21
  database server (ODB_SERVER) 2-3
  filling options 1-22
  Help 1-33
  LDM menu 1-4
  multiuser configuration 2-7
  preparing to work in 1-37
  product description 1-2
  query option 1-22
  single user environment 2-15
  task set 1-23
  top bar options 1-22
  windows 1-5
Cancel on a property sheet 1-31
Change Adrawing 10-11
Change Variable property sheet 9-17
Changing a component 7-4
Changing a reference assembly 7-25

Changing Assembly Structure 7-2
adding assembly by 7-3
adding component by 7-2
changing assembly by 7-3
changing component by 7-2
copying component by 7-2
removing assembly by 7-3
removing component by 7-3
Character, in variable name 9-10
Child component 1-8
Close reference 3-26
example 3-27
Command 1-33
cannot undo 1-36
help on 1-33
reversing 1-36
window access to 1-31
Command help 1-33
property sheet 1-33
Command palette
options 12-7
Component 1-7
adding 3-9
copying 7-12
mirroring 7-15
moving 8-24
obtaining information about 7-36
reordering 7-32
Component class 1-8
changing a component name and 7-6
Component geometry, creating relationship between 6-8
Component instance 1-7
changing 7-4
changing position in Assembly Structure 7-32
placing below another component instance 7-34
Component name 7-4
changing 7-4
creating a new class 7-6
Component orienting 8-32
Compressing a subassembly 7-30
example 7-30
Concurrent Assembly Mock-Up (CAMU)
features 1-2
Concurrent design 5-27
Configuring CAMU 2-7
choosing a host system 2-7
CVPATH prerequisites 2-10
DB_DAEMON_HOST 2-7
network consideration 2-8
Network File System (NFS) prerequisites 2-8
other system considerations 2-11
performance consideration 2-7
sample procedure 2-11
software prerequisites 2-10
testing multi-user configuration 2-13
troubleshooting 2-14
Constraint component 6-13
align 6-15
base component 6-17
fix 6-16
mate 6-15
orient 6-16
property sheet 6-14
Constraint deleting 6-20
example 6-20
Constraint in assembly 9-2
basic steps 9-2
change in model geometry 9-20
changing variable value 9-17
automatic mode 9-18
manual mode 9-19
component name 9-10
equation 9-12
exporting variable from active model 9-5
extended variable name 9-10
importing variable to assembly model 9-7
variable 9-2
variable name (imported component) 9-10
variable status 9-7
Context-sensitive menu 1-31
Copied Assembly 3-22
Copy Component property sheet 7-13
Copying a component 7-12
changing the copy 7-12
positioning 7-14
viewing and positioning 7-14
Create Attribute Group property sheet 11-4
Create Attribute property sheet 11-8
Creating
an assembly 3-2
handlepart 1-37
position constraint 6-7, 6-13
sample procedure 6-22
Creating an assembly
exporting by 3-31

Cursor moving, property sheet on 1-29
CV_ASSEMBLY_HANDLE_PART 2-3
CV_ASSEMBLY_NO_MODEL 2-5
CV_ASSEMBLY_PACKAGE 2-3
CV_MAX_ACTIVE_REF 2-5
CV_POSITION_CONSTRAINT_FEEDBACK 2-5
CV_UI_ACT_ASSY 2-4
CV_UI_DEFAULT_ENV 2-3
CVMAC program 11-19
  creating an executable 11-20
  query 11-19
CVPATH usage 2-20

D

DB daemon in .caddsrc file 1-38
DB_DAEMON_HOST 2-3
DDRAW 5-7
Default assembly package, setting 2-3
Delete Adrawing 10-12
Deleting
  a component 7-7
  a constraint 6-20
  a reference assembly 7-3
  a variable 9-23
Diamond-shaped button 1-30
Dimensioning
  model geometry 12-21
  the assembly 12-22
Dimensions
  regenerating 12-25
Disassociate Attribute Group property sheet 11-6
Display Structure property sheet 7-29
Display Unresolved Subassemblies 6-3
Documentation, printing from Portable Document Format (PDF) file 1-xx
Documenting the assembly 6-2, 12-2

E

Edit Attributes property sheet 11-11
Environments
  accessing from Assembly Design 1-23
  available from Activate New Assembly property sheet 3-3
  moving between 1-24
Equation 9-12
  constraint relationship in assembly 9-14
  listing 9-14
  solving 9-12
  status symbol 9-14
Exiting
  CADDS 4-5
  CAMU 4-5
Exploded view 10-6
  example 10-10
  selecting from within AD 10-9
  turning on and off 10-9
Export Variable property sheet 9-5
Exporting a reference assembly 3-35
Exporting an assembly 3-31
  active 3-35
  referencing with 3-33
  referencing without 3-32
Extended variable name 9-10
  example 9-11
External Assembly 3-21
  closing a reference assembly 4-3

F

FILE icon menu selection from 4-4
Filing 4-2
  active assembly 4-3
  active model in history replay mode 5-9
  active model or Adrawing only 4-7
  and exiting 4-4
  Assembly Structure 4-4
  creating a new model 4-6
  options 4-4
  removing implicit locks 4-3
  selected Adrawing or model 4-6
  solving positional constraints 4-2
  updating associative annotation 4-2
  while importing component 9-8
  without exiting 4-4
Fix 6-16
Fix Component property sheet 9-22
Flip option, using 6-8

G

Global Menu 1-24
Graphical mirroring 7-15

H

Handlepart 1-37
  adding components 3-8
  creating 1-37
Hidden Line Removal 12-3
  Remove Hidden Lines 12-5
  Restore Hidden Lines 12-5
HIDE OBJECT 12-3, 12-5
HIDE OBJECT interrupt 12-9
Highlight Component property sheet 8-44
Highlighting geometry 8-44
HLR generating using CADDShade II 12-17
Hybrid positioning 6-12

I

Import Component property sheet 9-8
Imported variable deleting 9-23
Instances adding 3-9
Interrupt HLR 12-9

L

Layer retaining assignment 8-23
Layer utilities, positioning a component 8-42
LDM menu 1-4
  activating new assembly from 3-2
Leaf component 1-8
List
  component data 7-35
  field on property sheet 1-29
  imported variable 9-7
  parametric positioning constraints 6-17
List assembly 7-39
List Attribute Groups property sheet 11-7
List Variable icon 9-7
Local Data Manager (LDM) menu
  accessing environments from 1-39
  illustration of 1-4
Locating a component 8-6
  with no associated model 3-8
Lock status, reference assembly in 5-34
Lock/Unlock Component/Model 1-24
Locking 5-31
  and viewing a component 5-32
  assembly structure 5-31
  crash recovery 2-24
  model 2-22
  property sheet 5-33
  root node 5-34
  what to 5-32

M

Managing data 2-21
Manual mode, solving equation 9-12
Mass Point 5-22
Mate 6-15
MDRAW 5-7
Menu, help 1-33
Mirror Component property sheet 7-16
Mirroring a component 7-15
  example 7-19
  graphical 7-15
  orientation specifying 7-23
  part 7-15
  procedure 7-16
Model 1-9
Model geometry 8-3
  activating 8-3
  creating 8-3
  creating guidelines for 8-2
  using source model 8-4
  viewing 8-18
Model Space Origin (MSO) 8-7
  locating 8-7
Modify Attribute property sheet 11-16
Modifying position constraint 6-17
  deleting 6-20
  offset value changing 6-19
  ways to 6-20
Move Component property sheet 8-24
  Orient Component option 8-29
  Rotate Component option 8-27
  six points orientation method 8-30
  Translate Component option 8-26
Move Component, on global menu 1-24
Moving a component 8-24
Moving between Adrawing and model 5-7
  example 5-10
Moving between environments 1-27
Multiuser CAMU 5-27

determining status 5-27
example of 5-28
path locking 6-6
positioning constraints 6-6
updating session 5-27
updating the assembly 5-27

N

Naming convention, variable 9-10
Node switching 5-7

O

ODB database server name 2-3
ODB_am_wh 2-3
ODB_SERVER 2-16
Crash recovery 2-16
multi-user environment 2-16
ODB administration tool 2-18
ODB_SERVER, crash recovery 2-24
Old assembly, activating 5-7
Online command help 1-33, 1-34
Open reference 3-26
example 3-27
Orient 6-16
Orientation, procedure 8-32
Orienting a component 8-32
example 8-35
Outline geometry 8-20

P

Parametric model, in Explicit environment 1-37
Parametric positioning 6-2
assembly constraint status 6-35
environment variable 2-4, 5-30, 6-18, 6-37
process order 6-4
tools 6-4
updating after modeling change 5-30, 6-37
updating automatically 6-4
Parent component 1-8
assigning a new parent 7-4
changing parent of reference assembly 7-25
Part mirroring 7-15
Permission read/write 5-31
Plotting
Assembly structure 12-29
part history tree 12-32
Popup menu in Assembly Structure window 1-31
Position constraint 6-7
Positioning a component
with no associated model 1-37
Positioning constraint
color scheme 6-21
creating 6-7
entity selection 6-9
multiple instances 6-10
solving 6-32
unique positioning 6-11
Positioning tools 8-42
for rotating a component 8-34
from Move Component property sheet 8-42
Print Component 7-36
Printing documentation from Portable Document Format (PDF) file 1-xx
Property sheet, using 1-28
Pushbutton 1-30
Pushpin icon 1-31

Q

Query
Adrawing 10-4
CAMU Status window 7-35
Component 7-36
Parametric Positioning Constraints 6-17
Reference Assembly 7-36
Querying ATB transactions 1-42
Quitting 4-8
selected Adrawings or models 4-5
suspended Adrawing 4-8
suspended model 4-8
the assembly 4-5

R

Read/write permission 5-31
Reference Assembly 3-21
controlling level 3-26
deactivating 4-3
exporting 3-35
listing 7-39

nested 3-24
Refreshing an assembly 5-36
behavior 5-36
structure and graphics 5-36
with multiple users 5-29
Refreshing an assembly, on global menu 1-24
Regenerate Component property sheet 9-20
Regenerating dimensions 12-25
Relation instance 1-8
in assembly structure 3-20
removing component using 7-9
Remove Attribute Group property sheet 11-18
Remove Attribute property sheet 11-17
Remove Component property sheet 7-7
Removing a component 7-7
all instances 7-7
component instance number 7-8
from the Graphics window 7-7
removing the relation by 7-9
selected instance 7-9
Removing a reference assembly 7-3
Reordering a component 7-32
above a component 7-32
above a reference assembly 7-33
below a component 7-34
Report created from attribute 11-19
Reset on a property sheet 1-31
Restoring Hidden Lines 12-15
Unblank Viewed Object 12-15
Reversing a command 1-36
Root component 1-7
Rotating a component 8-27
Runtime list 1-29

S

Save Restore
viewstates 10-12
Saving your work 4-2
Select Assemble 6-4
Selecting a component, property sheet 1-28
Selecting an option 1-28
Sibling 1-8
Solve Component property sheet 9-18
Solving equation, in assembly 9-12
Square button 1-30
Standard view 10-7
Subassembly
blanking 7-29
compressing 7-30
removing 7-7
reordering parent of 7-32
Suspended Adrawing 5-7
Suspended model 5-7
Suspended, when quitting 4-5
Suspending model, .caddsrc environment variable 1-38
Suspension buffer 5-9
Suspension buffer setting 2-4
Switching Node 5-7
Symbol, status of constraint equation 9-14

T

Template, Adrawing 5-4
TIM parts 1-40
Toggle Exploded View 10-9
Translating a Component 8-26

U

Undo AD Command 1-24, 1-36
Unexport Variable property sheet 9-5
Unfix Component property sheet 9-22
Unview component 8-18
property sheet 8-19
Updating
TIM parts 1-41
Updating assembly structure 4-4, 5-36
Updating parametric position 5-30
Updating session, multiuser CAMU in 5-27
User-defined information for the assembly 11-2
Using CVPATH 2-20
Utilities for positioning, Move Component property sheet 8-42
Utility construction planes, Move Component property sheet 8-42

V

Variable 9-4
applying change 9-18

changes to model geometry 9-20
changing value of 9-17
deleting 9-23
establishing relationship between 9-14
exporting from active model 9-4
forcing to remain constant 9-22
importing to assembly model 9-7
listing exportable 9-7
name 9-10
placing in the active assembly 9-7
preparing to write equation with 9-12
selecting to export 9-4
types 9-8
Variable anon 2-6
Verify object 7-36
Verify Object property sheet 7-37
Verifying an object 7-36
View component 8-18
property sheet 8-19
View Component property sheet 8-18
View state 10-5
DDRAW 5-7
MDRAW 5-7
Viewing a model
in the Explicit environment 1-37
outline geometry 8-20
with View Component 8-18
Viewing and positioning components
when component is created 3-12
Viewstates
restore 10-14
save 10-13
save restore 10-12
vp_links directory 2-23
crash recovery 2-24

W

Window 1-5
accessing functions from 1-28
Assembly Structure 1-5
Attribute Editor 11-11
CADDShade picture window 1-6
CAMU status bar 1-6
CAMU status window 1-6
command buffer 1-5
text window 1-5